BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

5.5.2.2. File System Tool

A general tool preferable in many embodiments is a file system with student access. In many instances, including elementary education, full view of and access to the system provided file system can be a disadvantage. Consequently this invention contemplates, providing a file system front end, or file system tool, that limits the student's view of and access to the system file system. This particular tool is advantageously implemented as part of the session manager and not as a material. With that implementation, customized iconic file representations are managed as part of the system area of the display.

One embodiment of such a file system tool presents a four level file system organized into shelves of books, each of which has sections, the sections having pages. A book representing a directory of files is opened when a student user selects its icon. The student navigates around the directory book by selecting tabs or by moving page by page, by selecting nextpage, lastpage, or other buttons. The student interacts with selected pages, or files, of the book, possibly creating new pages and sections. The student can also close the book. For most needs of elementary purposes, a single shelf with a few pre-specified books is adequate. More advanced students can be given permission to create and use multiple shelves with new books and to cut and paste pages and sections from book to book. A page of a book, a file, is preferably presented with the materials that process it. For example, user-created text or graphics pages appear with the word processor active. Homework and instruction pages appear with the appropriate materials.

5.5.2.3. The Scheduler/Calendar Tool

The schedule/calendar is an important tool and is preferably always present. It is accessed when the ABI system initiates materials to verify the student is permitted and scheduled for this material, and also invoked when the system terminates materials to schedule new materials. It is

accessed as a global function by the agent in response to a meta-request from the student seeking scheduling assistance. Further, it can be directly accessed by the student using the calendar tool icon appearing on the student desktop. When accessed, this tool displays a calendar book to the student, viewable in several ways.

This section describes, first, the schedule/calendar data, and second, the schedule/calendar processing. The schedule/calendar data is a subtype of the school-student data areas. Optionally, a student's personal schedule can be contained in the student data object. Generally, this data includes the following fields for each scheduled student activity:

Name of scheduled activity and optionally an identifying message;

Activity priority;

Deadline date and time, or a definition of a perpetual activity, which has periodic requirements but no completion date;

Link to material for the activity which in turn can specify activity completion criteria; for activity of a single student this is typically a particular instructional material; for group work activity a list of the students for the group and other communication information can also be in the calendar entry;

Activity characteristics, for example whether this was entered by the student or teacher and whether this is to be marked complete by the student or system;

Activity status, completion status and submission status of any required reports.

This exemplary data, sufficient to define a scheduled activity, can alternatively be entered by the teacher or by the student. If entered by the teacher, it can be protected from modification by the student. Data entry is preferably assisted by a teacher tool with a suitable screen format.

Optional activities can be entered by the student if the teacher permits.

The schedule/calendar tool can be directed by the teacher to permit the student a range of scheduling initiatives. These initiatives can range from permitting the student freedom to schedule materials as desired to requiring the schedule/calendar tool to enforce a fixed order of student activities. In a preferred embodiment this is accomplished by the teacher's specifying initiative parameters including a deadline date/time, D, and a criterion priority, P, in the school-student data. The schedule/calendar then schedules automatically all tasks with deadline less than or equal to D and with priority level greater than or equal to P. By varying D and P with respect to scheduled tasks the teacher can achieve the specified range of initiatives.

Schedule/calendar processing can be invoked by the executive software, by the student, or by the agent. As previously described, at the beginning of a student session or after termination of the use of a particular material, the executive software invokes schedule/calendar tool, first, to mark the terminated task complete, and then, to reference the calendar data in view of the initiative parameters to find activities requiring scheduling. If this tool finds only one activity in the calendar requiring scheduling, this required activity is initiated. If multiple required activities are found, the tool can, alternately, initiate the required activity of highest priority or allow the student a choice of which required task to initiate. If there are no required activities, the schedule/calendar tool allows student selection of the next task via the session manager.

The schedule/calendar tool can be invoked by the student by selecting the schedule/calendar icon in the system area of the display. When so invoked, the tool displays the student calendar and scheduled activities in various formats from a monthly overview to details of individual activities. If

permitted, the student can enter optional items or mark items complete.

The schedule/calendar tool can also be invoked by the agent when it receives a student meta-request of the type "What do I do next?" The agent retrieves the required and scheduled activities from this tool and also determines an expected time to complete each task based on student performance from the student data object and the average time required for each task from the materials header. In view of this combined information, the agent can present to the student an ordered list of activities scheduled according to their expected time to complete.

5.5.2.4. Communications Tools and Group Work Materials

In a preferred embodiment, the ABI system includes communication, or group work, materials integrated with the remainder of the system. As for other tools and materials, access to communications materials is granted by the scheduler/calendar tool. Communication work groups are assigned and scheduled in students' calendars with calendar entries preferably including the group members names and other communication parameters. When these materials are activated by the scheduler, the communication group is begun. Alternatively, students can spontaneously request the formation of a communication group by the selection of a communication material. The scheduler/calendar tool can permit group activation if the students have no other required activities. Alternatively, each particular communication material can also have specific access controls preferably set by the teacher that control the types of communication permitted and with whom the communications is permitted.

In manner similar to other tools and materials, the communication materials have an agent interface. Upon activation, they send initialization event messages to the agent specifying the global control variables they will be sensitive to, the educational paradigm adopted, and available

hints, helps, and other communication parameters. In alternative embodiments, the instructional materials interface standards include special categories for communication based work that enable the agent to control these materials with specificity. During communication work, these materials generate event messages at educationally significant points.

Thereby, communication materials are fully integrated into an ABI embodiment. Further, in a preferred embodiment, communication materials are implemented in a manner similar to other materials. First, each communication material has a particular communication task specific for that communication material or form of group work. The communication task manages the network interface for that particular type of communication or group activity by using the network protocols provided by the OS and ES, and provides its communication functions as global functions for access through an ABI system. Second, these functions are made available to the student in a manner similar to other materials through particular materials data that includes presentation items, sequencing logic referencing these global communication functions, and notations generating event messages for the agent. In an alternative embodiment, the communication materials can be programs, independent of the materials engine and perhaps part of the associated communication task, which internally generate the necessary agent event messages. In either embodiment, communication materials tasks can be written either in the ABI implementation languages, or in a special purpose communication scripting language.

The particular communication materials in a preferred embodiment provide forms of group work or communication including e-mail or message exchange, linking student groups for joint work on materials, and structured joint work such as contests with rules. Each of these forms of group work or communication is described in the remainder of this section.

A first form of group work implemented by communication materials is E-mail and newsgroups. These are useful for teachers to send information to their classes, such as schedule and materials changes and to communicate with absent students. Teachers can also exchange information with each other or obtain help for system and other issues. Students can use this form to obtain help and advice, especially from remote sources, communicate with their teachers, and share work or interests with other students. E-mail and newsgroups are easily incorporated as previously discussed. Then materials data is authored that grants access to these functions and generates appropriate agent event messages.

Student linking is another form of group work implemented by communication materials. Students in session at separate clients can link together for various exemplary activities including simply talking with each other by voice or text or for joint work on a particular material in which the students have either similar roles, as in developing a document using a word processor, or different roles, as in a simulation or game. Another activity of linked students includes group activities, in which position of participants within a virtual environment determines activity and role within activity. A final exemplary activity for linking student groups is moderated activity, in which participation is controlled by a special coordinating task that perhaps executes on a server system. An example of this latter activity is a spelling bee which is described in more detail subsequently.

In the preferred embodiment, student linking includes the following steps. The first step is identification of other students with whom a given student can link. The group can be defined by the teacher in the schedule/calendar entry for this activity, or alternatively, in a communication access control particular to this linked work activity. Second, links must be established between the students to be linked. These links can be to a single server communication materials task that receives and distributes messages.

Third, local and global actions must be determined. Local actions are those that result in output visible only to the user taking the action. Global actions are those that result in output visible to all the students in the linked group. These global actions can include communicating each student's input to all linked students, sharing information among all linked students, jointly creating information by students in the linked group, and storing jointly created information. The fourth step in linking is orderly disconnection from the linked group.

Linking can be implemented in alternative fashions. In all implementations the communications tools and materials for linked activities are integrated with the other components of an ABI system in one of the ways previously described. A simple implementation is to provide on each student's screen an icon and a message area for each linked student. Alternatively, one or more shared materials areas can be provided. Communication can be distributed through a single server task to which all linked students connect. A more advanced implementation of linking employs software packages similar to multi-user dungeons ("MUDs"), which contain a collection of virtual shared traversable spaces (called "rooms") in which users can interact either with each other or with elements within the room. MUDs are especially preferable for student group work in which different students have different roles with access to different system capabilities. For example, one student can be a recorder, having write-access to the group's notebook, while another can be the database expert, having access to a body of relevant data. MUDs are also be useful for teachers, communicating with each other within 'rooms' each set aside for a specific topic and forming a dynamic newsgroup.

Another form of group work implemented by communication materials is structured linking. Important examples of structured linking in which the students have different or structured roles are educational contests. Exemplary of such contests is a spelling bee, an exemplary embodiment of which

is described herein. In this embodiment, the spelling bee is managed by a server communication materials task, called the server spelling bee task, which preferably is executed on a server system, which communicates with local spelling bee tasks on the involved students' client system. Generally, the server spelling bee task initiates a spelling bee by accumulating connections with students local spelling bee tasks, then mediates the spelling bee by controlling student actions, and provides for orderly termination of the spelling bee. The local spelling bee tasks provide the communication functions accessed or required by the spelling bee materials data, which are scheduled or selected to invoke spelling bee participation, on the client systems. These materials also send event messages to the agent and are controlled by the student's agent. Alternately, the local spelling bee tasks can be programmed to communicate with the agent and perform the spelling bee without materials data.

In more detail, the spelling bee tasks carry out the following steps. The spelling bee server task is started at a teacher's request or automatically by the system. A list of eligible students can be selected by the teacher or automatically determined by the server task based on data in the student data objects, including class membership and a stated interest in spelling bees. The spelling bee activity can be scheduled for each individual student by the teacher or selected by the student. If a student desires to enter the spelling bee, the local spelling bee task is started by the ES and a message is returned to the server. No response within a specified amount of time is taken as indicating a desire not to join. If a student's data object shows that this is the first time in a spelling bee, an instructional and warm-up sequence can be authorized for the student by the agent in the server task. If enough eligible students join the spelling bee, the server task continues, otherwise it sends a termination message to those who have joined and terminates. Each local spelling bee task obtains space on the student display indicating the other players and their

inputs. Next the spelling bee begins, and the server task broadcasts the word to be spelled selected from a graded word list and the name of user to spell the word. Each local task echoes the word sent, preferably by requesting the on-screen agent voice the words as utterances with an appropriate affect. The spelling bee materials inform the student's local agent of the student's progress and performance in the spelling bee materials. The server task accepts input from designated user's local task and broadcasts it. The server task judges and reports on correctness of completed responses and, if incorrect, eliminates the student from further spelling requests. As student leave the spelling bee, the server task is notified and sends messages to the local tasks of continuing players in order to update their work spaces. In each case the student preferences for further spelling bees are checked and preference data in the student data object is updated. When only one student remains, the server spelling bee task terminates the game and reports results.

5.6. The Agent And The Agent Interface

The agent is an important component of this invention and is further described in this section with reference to the student data object, the agent interface, agent processing, and agent adaptivity. The agent comprises the student data object which contains data on the student's performance on the various materials and data on the student's pedagogic model. Other system components preferably have an interface to the agent in order that the agent can control the materials and guide the student in a uniform manner. Agent processing is divided into two phases, agent action processing and agent behavior processing. Finally, agent adaptivity in the preferred and in alternative embodiments is described.

5.6.1. Student Data Object

One student data object is created for each student in the ABI system and is a repository of data concerning that student. The student data comprises fixed data defining the student as well as evolving data describing the student's interaction with the system, the latter including at least current and past performance and data defining the agent's view of the student.

Preferably, the student data object is stored on portable media as a portable student data object under the student's control. In this embodiment, the portable student data object is plugged-in to a client system and the identity of the student if checked by access methods. Further, during system operation, each element of software of material that directly or indirectly, through the agent, seeks access to the portable student data object is validated and identified as previously described before access is granted. Alternatively, the student data object can be downloaded from server systems when a student access a client system. In this embodiment also, the software validation and identification is also advantageously performed. In either embodiment, the student is able to access any client system for provision of instruction from an ABI system.

In certain embodiments of an ABI system, a school or other educational provider can keep student related data in school-student data areas. This data includes information specific to a school, such as specific course or educational progress data, school schedules, etc., and is maintained in accord with the school's data retention and privacy policies. School-student data can be updated by messages generated by the agent and sent to school server systems. Also, the agent can consult this school-student data to guide its actions, for example, in response to scheduling meta-queries from the student. In alternative embodiment, such school-student data can be stored in the student data object. This can be advantageous when the student is associated with only one school, as in primary or secondary education.

For compactness of description and not by way of limitation, the following description of the content and operation of the student data object is according to such alternative embodiments. In the accompanying figures, data subtypes or fields that can be stored in school-student areas in appropriate embodiments is suffixed with "(S)." Data subtypes and fields that are preferably part of the student data object in all embodiments are suffixed with "(A)." In this subsection, this latter data is termed agent-specific data as it is closely related to the agent student model and agent processing.

Figs. 10A, 10B and 11 illustrate the structure and operation of the student data object. Figs. 10A and 10B conceptually illustrates an exemplary structure for student data object 1101, optionally containing school-student data as indicated. It is an object comprising structured data 1102 and methods 1103 for accessing and updating this data. Both agent-specific and school-student data is divided generally into global data 1104, materials related data 1105, including tool related data 1106, current lesson data 1107, and log data 1108.

Global data, defined to be information meaningful across ABI materials, includes such subtypes as system data, agent behavior preference data 1109, agent student model data 1110, and schedule data 1111. System data includes student identifiers, student passwords, access privileges, grade levels, class registrations, etc. Such information is typically school-student data. Both agent behavior preference and agent student model information is agent-specific data. Agent behavior preference data 1109 relates to the multi-modal behaviors generated by the agent and includes student defined preferences for these behaviors as well as a summary of past agent behaviors. Student preferences can include options relating to agent visual appearance - species, gender, dress, or perhaps, no visual appearance - and similar options relating to audio behavior and text production. The summary of past agent behaviors is

used to aid in the selection of reasonably varied future multi-modal behaviors. Agent student model data 1110 includes items modeling the student's persistent behavior which the agent uses to individualize its interactions with the student. Such data items can include material retention rate, hint effectiveness, and preferred rewards. Further data types in an agent student model can include measures of a student's intelligence, temperament, personality style, and so forth. Intelligence measures can be specified for multiple aspects of a student's personality, for example, measures according to the multiple intelligences of Howard Gardner. Personality style can include intuitive/formal preferences, such as right brain/left brain dominance. Updates to this data object made as the ABI client interacts with the student, and as the student matures, provide for agent adaptivity to the student. Finally, schedule data 1111 relates to assignments scheduled by the teacher or educational provider and unchangeable by the student, and is typically school-student data. Data for each schedule item can include due dates, reminder alarms, and priorities.

Each course in which the student is currently enrolled has a separate materials related data area 1105 for the materials providing instruction in this course, and includes both agent-specific and school-student data. In an exemplary embodiment, this latter data subtype includes standard and criteria data, usually set by the school system, which include objectives and standards the student must meet in the particular course, and milestone data establishing objectives already met by the student. Such information is typically school-student data. Information relating to a student's progress, performance, and use of tools is typically agent-specific data. Progress data includes data necessary for the student to leave the materials and resume the materials at the prior point. Performance data 1112 relates to student's performance over several lessons in the materials and can include mean performance, weighted moving averages of performance, patterns of performance, use of hints, use of

retries, and needed remediation. Using such performance data, for example, means and weighted moving averages, permits the agent to determine whether student performance is improving or declining. Tool data 1106 contains essentially similar but more abbreviated data about use of system tools such as the calculator, dictionary, and word processor. This data usually includes only milestones and performance information.

For example, in the case of a reading course, agent-specific data can include reading speed attained, vocabulary level achieved, and sentence complexity recognized.

The status of each lesson presented by instructional materials is accumulated in current lesson data 1107, which is preferably agent-specific data. This subtype is created upon lesson initiation and deleted upon lesson completion. In an exemplary embodiment, it includes short term measures of performance - such as error rates, weighted moving averages of error rates, and the use of hints and retries - short term measures of time latency - such as time to complete lesson segments and weighted moving averages of such times - work areas in which the agent can store information particular to the instructional materials - such as parameters to use in forming multimedia presentations - and lesson coaching parameters 1113. The lesson coaching parameters are used by the agent to provide feedback to the instructional materials so that their presentation can be individualized according to student performance. These parameters are governed by the instructional modalities employed by the particular instructional materials and can include values such as the seeding rate of new concepts, time pacing of the presentation, the density of examples, and the ratio of reinforcement.

Finally, the student data object has links to student log 1108. The log is preferably school-student data stored on server systems outside of the student data object. It is updated by messages generated by data object update methods called by the agent. Subject to school data retention and

privacy policies, the log can store all messages input to agent processing and all actions from agent behavior processing. Preferably, this data is used to generate reports for teachers, administrators, and parents. It is also of use to instructional materials authors and educational researchers, who can use the detailed audits of student behavior and system responses available in the log in order to improve instructional materials or to develop new modalities of instruction.

With its carefully partitioned and functionally defined interfaces, the ABI system is easily adaptable to new modalities of instruction as well as to merely installing new materials.

The student data object also includes access, data updating methods, and data query methods. The methods are advantageously grouped in particular application programming interfaces ("API") according to their function of type of data accessed. As previously described, methods of the access API validate the student attempting to access a portable data object and validate and identify software prior to its generating any student data object requests. An exemplary updating method 1114 includes two components, triggering event type 1115 and action list 1116. When a validated and identified system component requests agent processing of updates to the student data object, the agent sends a message to the object including an update event type and a list of relevant parameters. The updating methods are searched to find one with a triggering event type which matches the event type of the update event. The one or more methods having matching event types are then executed by performing all the actions in the included action list using the parameters in the update event message. Each action uses the supplied parameters to update data elements in the student data object. Preliminary to any update, it is advantageous for the update methods to execute rules to ensure the update is generally reasonable, by, for example, not occurring too frequently. A plurality of actions is

generally associated with each method because it is typical for one event to cause changes in several student model data elements. For example, an update event related to the use of a tool causes changes in the relevant tool data subtype as well as the associated instructional material subtype. Further, when a lesson is completed, the methods executed appropriately summarize student data from the current lesson subtype into all the permanent data subtypes.

For a further example, at lesson completion parameters such as 'hint effectiveness' in the agent student model data 1110 are also updated. The 'hint effectiveness' parameter is accessed for uses including agent decision weighting. One of the methods associated with the event type 'end of lesson' updates the parameter 'hint effectiveness' in the following exemplary manner. The performance parameter 1120 'use of hints' is accessed to determine if update is required. If hints were provided to the student, the 'current lesson log' 1108 is used in calculating the two components of the parameter 'hint effectiveness' - 'hint effectiveness-before-first-try' and 'hint effectiveness-after-first-try'. These two components each hold an array of values which are used to compute a weighted moving average. In an alternative embodiment, a more complex formulation of this parameter can be used to provide a more detailed analysis of hint effectiveness - each of its two components can be separated into sub-components corresponding to subject area, hint types and other instructionally useful measures.

It is further preferable, for data elements comparing a particular student to class peers according to various measures be entered into the student data object. This can be done by the agent executing appropriate student data object update methods in response to messages containing such student comparisons generated from school-student data on school server systems.

Fig. 11 is an exemplary illustration of how a typical student action event updates the student data object. Update message 1201 is sent 1206 to the student data object from a

component of agent processing to cause student model update. The update event type is "exercise_done" and exemplary associated parameters are as indicated at 1202. In this case method 1203 with triggering event type "exercise_done" is executed by performing the four associated actions. The first action updates relevant data elements in tool data subtype 1106. The second action updates the log 1108, by, preferably, sending to server systems an information message describing this event. The log is typically updated for all events. The third action updates timing latency data elements 1205 in current lesson subtype 1107. Finally, the fourth action updates student performance data elements 1204 in current lesson subtype 1107.

A validated and identified component of the ABI system needing to determine the value of particular data element in the student data object does so by sending an query message to the student data object requesting the desired data element. The inquiry method for that data element retrieves and returns the desired value. Such inquiries are primarily made by the agent on the student client system. In the preferred embodiment in which school-student data is stored apart from the student data object on server systems, such queries are performed directly by inquiry and report generating programs.

Further aspects of student data object operation include backup of portable student data objects and student data object initialization. For reliable security and integrity, especially in case of loss or physical damage, it is advantageous that portable student data objects be backed up on a backup system. Further, it is preferably that such backup preserve the privacy of information in the portable student data object. Accordingly, in the case of portable media such as hard or floppy disks, a copy of the already encrypted data is sent to the backup system for storage. In the case of smart-card-like portable media, an interface can be provided which causes the stored data to be externalized as an encrypted stream for backup storage. Backup are

preferably performed with sufficient granularity, such as at educationally significant events and at local session termination. Centrally stored student data objects are backed up according to the facilities of the object-oriented database in which they are stored.

Further, it is advantageous that a new student data object be initialized. Thereby, instructional materials can have the advantage of an agent already knowledgeable about the student, and the time consuming and perhaps tedious process of initial agent model construction can be greatly shortened. During instruction, the initial model is refined and sharpened as previously described. Such initialization can be performed in numerous ways. First, the student personally, or a teacher, parent, or other knowledgeable person for the student, can directly enter estimates of parameters used in the agent's pedagogic or cognitive model of the student. Second, model parameters can be derived from tests administered to the student prior to any instruction. Such tests can include those specially created for student data object initialization as well as standard tests of a student's abilities. These latter include IQ tests for student abilities, ABLE™ tests for adult abilities, Thorndike Tests of Temperament (TDOT™) for personality profiles, and so forth. Specific IQ tests to which this invention is adaptable include the Stanford-Binet™ and the Wechsler™ tests.

Specific initialization can optionally be provided for certain specific student data object fields. Reading skills can be measured and initialized according to one of the many standard reading and vocabulary tests known in the educational arts. Multiple intelligence traits, right brain/left brain dominance, and so forth, can be manually initialized by a person who is knowledgeable about the student, or who has carried out a personality assessment of the student.

5.6.2. The Agent/Materials Interface

The structure of the interface between the agent and the materials is important in the ABI system. It permits a single agent to control a wide range of materials through which it guides a single student. The agent achieves this by advantageously maintaining a model of the student's pedagogic characteristics, which it references in diverse situations to determine its actions. This section, first, describes the general procedural structure of this interface, and second, describes the preferred model for the content of the interface. This preferred model is structured according to the instructional material interface standard (herein called "IMIS")

Communications between the agent and the materials is bi-directional. Events containing parameters are sent to the agent by the materials at educationally significant occurrences. In response, the agent sets global parameters controlling the materials and returns messages confirming actions proposed by the materials. In circumstances in which the materials needs to coordinate displays with the agent, it communicates synchronously with the agent. For example, when the student requests help or a hint, the materials can need to synchronously obtain the agent's permission to offer the help or hint. In other circumstances, the materials can asynchronously send informational messages to the agent. Such asynchronous agent input and possible output can give the system the appearance of spontaneity.

The agent/materials interface can be implemented in any convenient manner in a given OS. For example, it can be built on explicit messaging, shared memory areas, procedure calls to a socket interface, or other technology.

The global parameters set by the agent and which control the materials are preferably state variables that the materials sequencing logic references in order to make educationally significant sequencing decisions. The meanings of state variables to which a particular material is sensitive can be established at materials initialization

according to specifications in a header materials data entry. Examples of such variables range from simple flags, such as those controlling the availability of helps and hints, to more sophisticated parameters, such as those controlling the rate of new concept introduction, the density of examples, or the speed of discrimination exercises.

Communications from the materials to the agent are controlled by notations in the materials data. A notation includes an event type, parameters associated with the event, and the condition under which the event is constructed and sent. Notations are activated when they are encountered in sequencing logic in the materials data. There are two classes of notations, "point-wise" notations and "toggled" notations. Point-wise notations are evaluated exactly once when they are activated. Toggled notations are evaluated at each input event following their activation until they are inactivated by a later notation.

Notations vary according to the materials. Some materials, such as simple e-mail, can contain no notations. Tool materials can contain notations indicating only correct or incorrect use of the tool. Most instructional materials data contain several types of notations. Generally, the events generated by these notations send information similar to the following: number of retries, measures of rate in fluency drills, measures of performance such as percent correct and the number of tries, partition of exercise items into similarity sets, and measures of the relative difficulty of items.

Table 2C illustrates exemplary types of notations generated by typical instructional materials.

TABLE 2C: TYPES OF NOTATIONS

NOTATION TYPE	WHEN GENERATED	TYPICAL PARAMETERS
POINT-WISE NOTATIONS:		
Lesson initialization	At start of lesson	Educational paradigm used; meanings of state variables
Partial response	At partial answer for a requested input item	Correctness of user input, time required
Full response	At complete answer for a requesting input item	Name of exercise, difficulty, related exercises; correctness of user input, time required,
Help/hint request	User request for help or hint	None
End of lesson	End of lesson	Objectives satisfied by lesson; performance; time required
Point	At an error	Screen location that agent can highlight or move to
Say	At an error	Message that agent can produce
TOGGLED NOTATIONS:		
Latency	No input event	Elapsed time, expected type of input
Inappropriate input	Inappropriate inputs from student	Number and type of inappropriate inputs

5.6.2.1. The Agent/Materials Interface Standard

In order that the student's agent can act generally to provide student guidance and control material presentation in a manner individualized to the student's pedagogic characteristics, it is preferable that an embodiment of this invention use an instructional materials interface standard (herein called "IMIS"). According to IMIS, it is preferable that the detailed content generated by the notations and passed in messages to the agent be structured in a standard fashion according to the particular educational paradigm adopted by the materials and independent of the particular content of the materials. Further, it is preferable that the

materials adopt one of a limited and defined set of educational paradigms contemplated in the standard. Finally, it is further preferable that the rules referenced by the agent in its associated processing tables and performance data in the student data object be structured according to the standards of the IMIS. Thereby, IMIS provides the agent with a materials independent view of the student performance.

IMIS is not limited to a particular set of educational paradigms. Any standard set or sets of paradigms appropriate to the intended students can be adopted for the interface standard. It is preferable that the standards adopted be based on principles of educational psychology and sound educational practice. In the following, this invention is described according to an IMIS appropriate for elementary education. The paradigms described below are not limiting.

Exemplary educational paradigms

Exemplary educational paradigms, also known as modes of instruction, are listed in Table 3.

TABLE 3: EXEMPLARY EDUCATIONAL PARADIGMS

Interactive tutorial
Fluency exercise
Paired association exercise
Discrimination formation exercise
Simulation exercise

Each of these educational paradigms is preferably handled differently by the agent in response to differing descriptive information and student performance data. For example, a sequence of correct responses in a fluency exercise is expected. On the other hand, a sequence of correct responses in a paired associates exercise can be worth while for the agent to comment on.

Exemplary paradigm classification

The exemplary embodiment of IMIS standardizes these educational paradigms according to three pieces of information: the instructional context, the instructional format, and most specifically, the subject area. Materials notations should preferably specify all pieces for maximum agent flexibility, although the ABI system is adaptable to the materials specifying any number or none. If none are specified, agent actions are independent of the educational paradigm.

The instructional context is the specific mode of instruction being presented to the student by the materials. Examples of instructional contexts are:

TABLE 4: EXEMPLARY INSTRUCTIONAL CONTEXTS

Prerequisite
Test
Review
Pretest
New material
Introduction
Discrimination
Review
Practice
Fluency exercise
Review
Unit mastery test

Within each instructional context, materials can adopt instructional formats, the second component of the IMIS specification. Examples of instructional formats are:

TABLE 5: EXEMPLARY INSTRUCTIONAL FORMATS

Multiple Choice
Fill-in-the-blank
Unprompted
Selected from list
Paired Associates
By letter
By dragging
By matching
Computations
Simulations
Identify parts of figures or text
Simulation games

Finally, student performance should preferably be stored relative to the subject area being worked on, as necessary for course level reporting. Thus, the third component of the exemplary IMIS is the subject area, such as mathematics or reading.

Exemplary standardization of agent data

IMIS standardizes system tables and data relevant to agent action processing according to this triple of information -- instructional context, instructional format, subject area -- which characterize the education paradigm adopted by the materials. The following description discusses standardization of notations in the materials, standardization of data in the student data object, and standardization of the agent action processing tables.

Notations in the materials are standardized according to IMIS as follows. For a lesson, exercise, or item, each material adopts one of the standard set of educational paradigms, or modes of instruction. The parameters to be passed to the agent in an event message are determined by the instructional context and each instructional format of this paradigm.

The following table contains an exemplary sequence of notations for a "prerequisites review" education paradigm,and

the parameters relevant to agent action processing that each notation sends to the agent.

TABLE 6: EXEMPLARY NOTATIONS FOR PREREQUISITES REVIEW

Site of Notation	Data Sent to Agent
Instructional Sequence Entry Header	Context Type: Prerequisites Review Unit Name: Selecting Gender Pronouns Subject Area: Grammar Average Time: 5 minutes Number of Exercises: 8
Exercise 1 Entry Header	Format Type: Multiple Choice Number of Options: 2 Difficulty: 0.3 Hint: Available
. . .	. . .
Exercise k Entry Header	Format Type: Fill in Blank Subformat: Select from Words in Previous Sentence Difficulty: 0.5

An aspect of these notations is to provide information about the quality of the student response, which, in this example, is given by the *a priori* probability of a correct answer. For example, selecting the correct response from two choices has an *a prior* probability of 0.5 if the choice is made at random. Filling in a blank correctly is less likely by chance, if no cuing is provided in the exercise.

Advantageously, notations containing the parameters associated with a given instructional context and a given instructional format can be stored as templates in libraries. Notations are available in these libraries to generate necessary messages at materials initialization and during materials processing. An instructional designer then need only consult these libraries to obtain the notations appropriate to the educational paradigm of the materials being authored.

The standardization of the student data object according to this exemplary embodiment of IMIS for elementary education is simply achieved by storing student performance data according to instructional context, instructional format, and subject area. Thereby, these characteristics can be taken into account when comparing student pedagogic performance in general across subject matter areas.

The final component of IMIS standardization is that of the agent action processing tables, the policy filter table, the decision weight table, and the selection rules. Event messages from the materials inform the agent of current values for the instructional context and instructional format. Since these values are parameters available to evaluate the conditions and functions contained in these tables, these tables can be, in effect, segmented into parts each corresponding to a particular instructional context and instructional format. Since there is considerable overlap between the parts of these tables, the number of rules does not proliferate. The current subject area is also available to segment the tables in the cases of those subjects that can require special treatment by agent action processing. Further, the IMIS standardization permits a more systematic and effective use of the mechanism which the agent uses to set global variables in the materials. These variables facilitate adaptive adjustments of instructional parameters, such as seeding rate and amount of prompting. These variables can be more effectively set in view of the current educational paradigm as indicated by the current values of the instructional context and format. IMIS is also useful in providing information to the student in response to "Where am I?" inquiries. The system can use the information contained in the entities in the information triple to respond, "You are halfway through the drill and practice on fractions," for example.

Alternative agent embodiment enabled by IMIs

In alternative embodiments of agent action processing, the segmenting of the policy filter table and decision weight table, which provides a simplified 'intelligent instructional agent' for each kind of instructional context, instructional format, can be augmented by other techniques from artificial intelligence. Thus, the customized rules and functions contained in the action table can be augmented software modules, which extend the agent and are constructed based on such techniques as production rules systems with rule propagation or neural nets. Such additional modules could, for example, find complex patterns of student error unanticipated by the instructional designer. As additional artificial intelligence methods are incorporated into the agent software, new materials notations can be added to the notation repertoire.

5.6.3. Agent Action Processing

Agent processing is divided into agent action processing, which determines on-screen agent display actions in response to input events, and agent behavior processing, which transforms display actions into displays to the student. In a preferred embodiment, agent action processing is rule based and event driven, and agent behavior processing is table based. This and the succeeding section describe the preferred embodiments of these divisions of agent processing.

Fig. 8 illustrates in more detail agent action processing. This processing is activated when event messages 801, representing either input events sent from the I/O handlers or educationally meaningful messages from the materials, are sent to the agent software. It transforms the input event messages into lists of display actions that activate the agent display and also has the important side effects of updating student data object 806, as indicated by arrow 815, and of setting materials control parameters, as indicated by arrow 816. It is possible a particular input

event can generate all or none of these outputs and side effects.

Agent action processing proceeds through three steps: event filtering, candidate action weighting, and display action selection. In common, each of these steps references rules in an associated table of rules. These rules include relations, functions, and procedures, all of which reference input parameters. Input parameters can be any appropriate data carried in the input event message and any appropriate data stored in student data object 806. In particular, as indicated by arrow 815, fields from the following subtypes in the student data model are referenced: the current lesson subtype, the materials specific subtypes, and the agent pedagogic student model subtype. In alternative embodiments where the agent interface and data is structured according to an IMIS standard, event input parameters include parameters characterizing the current educational paradigm adopted by the materials. These parameters can be used to select rules applicable only to this paradigm for reference and activation.

Turning now to the steps of agent action processing, the first step is event filtering 807, referencing policy filter table 803. This step is described with reference to Table 7, an exemplary policy filter table.

TABLE 7: POLICY FILTER TABLE

	POLICY FILTER TABLE
Active	Rule
Y	B(p_1, p_2, ... ,p_n),(type, subtype, x_1, x_2, ..., x_n)

Y	(item_diff>LIM),(congratulation,diff,2, ...)
Y	(rate>f*average),(congratulation,rate,k, ...)
Y	(exercise_done),(congratulation,done,r,t, ...)
N	(latency > m),(hint, constructed, ...)
N	((error_rate<k) & (WMA>MAX)),(remediation, review)

The first row illustrates a generic policy filter rule, while subsequent rows illustrate exemplary filter rules. The generic rule has a condition and a consequent. The condition is Boolean expression, B(), of one or more conditions, p_n, each of which is a function of the available parameters. Default parameters in a rule can be overwritten by parameters from the input event. The consequent is a list of an agent action type, "type," an agent action subtype, "subtype," and zero or more parameters, x_n. The type is a major mode of agent behavior, for example, "congratulate the student," and the subtype modifies that behavior, for example, "because of the student's rate." The parameters provide additional information to be used in constructing an utterance, for example the time that the student took to complete an item or the name of the item. Each rule can be marked active or inactive, possibly as a result of selection during materials initialization or as a result of selection according to IMIS educational paradigm parameters. Processing a filter rule consists, first, in evaluating each active boolean, and second, for each such active and true expression, placing the consequent and its parameters into the list of candidate actions. The list of candidate actions is then passed on to

action weighing 808, the next processing step. Also in the student data object, the current lesson subtype and materials specific data subtype are updated, if necessary, with data from the input event message. For example, upon item completion, performance results need to be updated.

Action weighting 808 references decision weight table 804 and assigns a numeric weight to each action in the input candidate action list. This step is described with reference to Table 8 an exemplary decision weight table.

TABLE 8: DECISION WEIGHT TABLE

DECISION WEIGHT TABLE	
Active	**Rule**
Y	(type, subtype), weight_function(x_1, x_2, ..., x_n)

Y	(congratulation, diff), weight_function_j(Y_1, Y_2, Y_3)
Y	(congratulation, done), weight_function_k(Z_1, Z_2)

The first row illustrates a generic policy filter rule, while subsequent rows illustrate exemplary filter rules. The generic rule has an index and a weight function. The index is a type and subtype pair, and the consequent is a weight function of zero or more parameters, x_n, with a value between 0 and 1. Each rule can also be marked active or inactive, possibly as a result of selection during materials initialization or as a result of selection according to IMIS educational paradigm parameters. Processing the list of candidate actions consists of, for each candidate action, selecting the weighting rules that have a type and a subtype matching that of the action and then assigning a relative priority between zero and one to the action as determined by applying the weight function to the available parameters. In this manner all the candidate actions on the candidate action list are assigned a numerical weight. The list of weighted

candidate actions is then passed on to action selection 809, the last processing step.

Action selection 809 references selection criteria table 805 to select a final set of actions from the input candidate action list. This step is described with reference to Table 9, an exemplary selection criteria table.

TABLE 9: SELECTION CRITERIA TABLE

SELECTION CRITERIA TABLE	
Active	**Rule**
N	(criterion)

N	(use all actions with weight > MIN)
Y	(use the most highly weighted action)

The selection criteria table consists of a list of available methods for the candidate action selection process. A selection criterion uses the computed weight and perhaps other parameters to select one or more final actions from the candidate weighted actions. Exemplary selection criteria are illustrated in the last two rows. It is anticipated that one criteria is marked active as a result of selection during materials initialization or as a result of selection according to IMIS educational paradigm parameters. The active criteria is applied to select zero or more final actions. Some final actions are executed locally to provide control to the materials by setting control parameters, as indicated by arrow 816. Other final actions can cause update of fields on the student data object, in particular in the agent pedagogic model subtype, as indicated by arrow 815. The remaining final actions are display actions which are passed to agent behavior processing in display action list 802. The student data object is updated to record the selected display actions.

Action processing is illustrated with reference to the exemplary rules in Tables 7, 8, and 9 by considering the results of processing two input events, one of type "item done" with a "difficulty greater that LIM" and one of type "exercise done." Applying the exemplary filter rules from Table 7 results in two candidate actions, one being (congratulation, diff) and the other being (congratulations, done). No other exemplary boolean expression evaluates to true. Next the exemplary weighting rules from Table 8 are simply applied by selecting and evaluating the correct weighting function. The resulting exemplary candidate action weights are taken to be 0.8 for (congratulation, diff) and 0.5 for (congratulations, done). Finally, the one active exemplary selection rule from Table 9 is applied which selects the one final action (congratulation, diff).

The preferred embodiment here illustrated involved no rule propagation. In alternative embodiments more sophisticated artificial intelligence programming techniques could be employed as is apparent to those of skill in the art. For example, rule propagation and general production rule systems could be used to transform events to actions. Other applicable techniques could involve neural nets, predicate logic, and so forth.

5.6.4. Agent Behavior Processing

Agent behavior processing transforms display actions into output displays to the student. In a preferred embodiment, agent behavior processing selects from tables of possible display behaviors based on the input display action list and its parameters and on fields in the student data object describing student preferences and recent agent display behaviors. Further, in a preferred embodiment, agent behavior processing generates either display scripts which are sent to the I/O handler, or scripts with associated data snips of sound and animation that are embedded in applets. In alternative embodiments, intermediate scripts are not used and behavior processing generates I/O handler commands in

real time. This section describes agent behavior processing, first, with respect to the table data structures used, and second, with respect to the processing steps.

Generally, agent behavior processing includes two steps. First, the message and its emotional affect, which is to be displayed or spoken by the on-screen agent, called herein the "utterance," is selected according to the input display actions. Second, the utterance and affect are sent to the preferred persona object where it is integrated into a complete display, including animation, graphics and audio. The display is preferably represented as a script which is then sent to the I/O handlers. In behavior processing, the selected affect in an important parameter. The affect further characterizes the intent of the utterance. For example, an utterance of a "congratulations" type is associated with a positive affect that can range from "happy," in cases of superior performance, to "encourage," in cases of improving but still less than average performance. The display behavior of the on-screen agent, preferably represented as one or more personae interacting with each other and the student, is strongly responsive to the affect parameter. As is known in the arts of film animation, infusing an animation with an affect or emotion gives resulting images a human and life-like quality. Such a quality is important so that the virtual tutor aspect of the ABI system engage the student in order to improve instructional results. Also, by storing recent affects selected in the student data object, as well as student preferences for such parameters as affect intensity and presentation, agent behavior processing can present a coherent and life-like image agreeable to the student over one instructional session or over several instructional sessions.

Affects are associated with and further characterize the intent of utterances. They are selected as part of utterance generation processing for use in visual display generation processing. Utterance generation processing depends on the

type, subtype, and parameters of the current event as well as in the student data object, which contains the agent student model, student pedagogic parameters, and recently generated behaviors. Therefore, preferably, the selected affect and generated visual displays are responsive to the student pedagogic characteristics or, alternatively, the student cognitive style.

Turning details of utterance generation, utterances and associated affect are generated by selecting options from a hierarchy of tables. This hierarchy, having typically many choices for each level, provides variety in agent behavior. At the highest level of this hierarchy are the tables of utterance templates 908, which are indexed by action type and action subtype. Typically, there are many utterance templates and associated affect for any given action type and subtype, and indexing is according to additional parameters that occur in action lists and additional parameters of student characteristics from the student data object. For example, utterances and affects are further selected according to the educational paradigm employed, student grade level, and student preferences.

At the next level of the utterance generation hierarchy are tables of named slots, which are parametrized phrases, or in other words, functions which use input parameters to return phrases. Utterance templates contain named slots as well as invariable words and sounds. When an utterance template is evaluated, slots with the specified name are selected from the slot table. Typically, there are many slots of a given name, which are further indexed according to additional action list parameters and user characteristics from the student data object. When a slot is selected for use in the utterance template, slot parameters are passed from the utterance template. Alternative embodiments of this invention are adaptable to hierarchies with additional levels of utterance generation tables or additional types of tables at a given level in the hierarchy.

Exemplary Table 10 is a small segment of the utterance template table for an action type of "congratulations" and an action subtype of "performance" and appropriate for a test type paradigm. Table 10 illustrates that the student grade level, obtained from the student data object, is an additional indexing parameter for these utterance templates. Further, each illustrated template includes a slot named "performance-streak-slot," with parameters of grade level, number of problems answered correctly, and total number of problems.

TABLE 10: EXEMPLARY UTTERANCE TEMPLATES

(congratulations, performance, grade=4, "You got <performance-streak-slot(4, numright, total)> right.")
(congratulations, performance, grade=4, "You are really getting the hang of it.")
(congratulations, performance, grade=8, "Quite a winning streak - <performance-streak-slot(8, numright, total)>.")
(congratulations, performance, grade=12, "A stellar performance getting <performance-streak-slot(12, numright, total)> right.")

It is preferable that the utterance template table have a large number of utterance templates appropriate to a large number of educations situations. Since the preferable number of type and subtype combinations is at least approximately 500, and since it is preferable to have 10 to 100 responsive utterances per pair, a preferable utterance template table has at least 25,000 entries. A less preferable table has at least 5,000 entries and a more preferable table has at least 50,000 entries. In an embodiment of the invention, less used utterances have fewer candidate responsive utterances without impairing the image of system spontaneity. Thus this invention is adaptable to utterance tables of 1,000 entries.

Exemplary Table 11 is a small segment of the slot table for slots named "performance-streak-slot," any of which could be employed in the previous templates. The illustrated slots

evaluate their input parameters to results in phrases. For example the last slot results in the phrase "90 percent" if numright is 9 and total is 10. Further, slots can contain a condition which controls their applicability. In this example it is the fifth component of each entry and only the first slot has an non-null condition.

TABLE 11: EXEMPLARY SLOT ENTRIES

(performance-streak-slot, 4-12, numright, total, <numright> > 0.8*<total>, "Nearly all")
(performance-streak-slot, 8-12, numright, total, nil, "<numright> out of <total>")
(performance-streak-slot, 8-12, numright, total, nil, "100*<numright>/<total> percent")

Slots are used to create final, generated utterances with considerable variety. Therefore, it is preferable that the slot tables have at least the same number of entries as the utterance template table. A preferable slot has at least 25,000 entries. A less preferable table has at least 5,000 entries, and a more preferable table has at least 50,000 entries.

The final display of the preferred on-screen agent object is generated from Display Behavior Tables 904 with a similar hierarchical table data structure to that used in utterance generation. At the highest level are on-screen agent actions which contain a cast of one or more personae. At the next level are persona types which the student selects once, or at most once per session. At the next level, associated with each persona type is a library of display behaviors indexed by affect. Exemplary affect types include the following: sad, objective, pleased, happy, disappointed, announce, remind, encourage, reinforce, model, prompt, hint, joke, and tutor. Typically, each affect has many possible behaviors and these are further indexed, as for utterance generation, by parameters appearing in the action list and the student data object. The display behaviors are

structured as scripts containing named display objects. These named display objects can optionally involve voice, audio, graphics, or video displays, and they are contained in scripts which can optionally specify a timed animated display or a branching display, where the branches are dependent on student reactions to the display. At the lowest level in the preferred embodiment are the individual named display objects. As for slots, typically there are several parametrized instantiations of each named object. These instantiations are indexed according to the same parameters indexing the display behaviors and in turn use these parameters to generate displays. Alternative embodiments of this invention are adaptable to hierarchies with additional levels of tables for display generation tables or additional types of tables at a given level in the hierarchy.

Exemplary Table 12 is a small segment with personae types adapted to elementary education.

TABLE 12: EXEMPLARY ON-SCREEN AGENT ACTIONS

A frog persona writing the congratulatory message on a tiny blackboard, then jumping up and down croaking 'Hurray'.
A robot persona creakily waving its arms and saying the congratulatory message.
An animal persona wearing a cheerleader's outfit jumping up and down, chanting the congratulatory message as a cheer.
The cat says the general part of the encouragement message while pointing to his bird sidekick. The bird then flies across the screen while saying the specific part of the message.

It is preferable that the on-screen agent have a richly varied and engaging behavior. The persona types preferably include recognizable popular figures as well as a rich variety of specially created figures. Persona type tables have a preferable selection of at least 100, and less preferably at least 50 persona types. The libraries of display behaviors preferably have many behaviors for each affect. Since the preferable number of affects is at least

50, and since it is preferable to have 50 to 100 responsive behaviors per affect, a preferable behavior libraries have at least 2,500 entries, and more preferably 5,000 entries, per persona type. A fully implemented ABI system can have preferable behavior libraries with approximately 125,000 entries. This number can be less in situations were more popular persona have fully configured behavior libraries while less popular persona have more limited libraries. Further, the number of persona types can be advantageously limited to only the most popular or more appropriate for the type of student. Thus this invention is adaptable to libraries with approximately 1,000 entries of responsive behaviors.

In implementations of this invention, it is important that the actual content of the utterance tables and the display tables and libraries be sufficiently creative and current to engage the student. Therefore, this content is preferably created by artists, animators, writers, and other creative talent. These elements of sound, voice, animation, graphics and video are collected into libraries of data snips and stored in archives. Further, it is preferable that these tables have an extensive and varied content in order that agent displays repeat only infrequently.

Turning now to the process of generating a complete agent display behavior, the steps of this process are described with reference to Fig. 9. The first step is utterance generation 906 which receives as input action list 901. Utterance generation indexes the utterance template table in accord with action list parameters, in particular action type and subtype, and student preferences from the student data object, to obtain candidate utterance templates responsive to the current situation. If no candidates are found, the student preferences are relaxed in a preset order until at least one candidate is found. Typically, more than one candidate is found, and then one template is selected at random from among those templates which have not yet been used previously in this session. A record of agent display

behavior during the session is stored in the student data object. The named slots of the selected template are resolved in a similar fashion, by finding candidate named slots and selecting one at random that has not been recently used. An utterance is then generated from the template, the named slot, and the input action list parameters. This utterance and its associated affect is passed to visual display generation.

The second step of agent behavior processing, visual display generation 907, uses the input utterance and associated affect to select candidate responsive displays of that affect from the libraries associated with the student's current persona(e) preference contained in the student data object 905. As with utterance generation, if no candidate is found, student preferences are relaxed until one is found. Typically, more than one is found, and one is selected randomly avoiding recent selections in this session. Display objects referenced in the selected display are similarly found. Finally, the utterance message is sent to the selected display and display objects. These objects contain performance methods that are called to generate and output display script. In a preferred embodiment, the script with references to data snips 909 of voice, sound, graphics, video and animation is incorporated into an applet. An exemplary applet can be a program fragment representing a character scratching his head while saying the utterance and moving across the screen. This output applet is sent to the I/O handlers in the ES for student display 908. Since this completes agent processing for this student or timer event, the system then waits for the next action at wait point 902.

A display action can also reference a pre-formatted animated sequence stored in the data snip library 909 in which voice, sound, graphics, animation and video have been already integrated. In this case, portions of utterance processing 906 and visual display processing 907 can be bypassed.

In alternative embodiments, selection of utterance or persona candidates from the available contents can be done other than randomly. For example, agent behavior data saved during a session in the student data object can be used to construct performances extending across instances of student actions. In particular, utterance templates and display objects can be selected in a connected manner in order to give agent display the appearance of continuity and realism. This invention is adaptable to other methods of generating performances parametrized by affect and utterance that are known in the art. These methods can include advanced text to speech techniques, 2-D and 3-D graphical generation, and VR effects.

The visualization of agent actions can involve a single character or several characters interacting with each other. If multiple characters are used, they are preferably interrelated. A typical cast is a cat with a bird sidekick. An exemplary interaction of these associated characters involving both dialogue and movement is presented in Table 12.

Continuing story line scripts can be available from the ABI system on a daily basis in a manner similar to a continuing plot in a daily comic strip. These story lines can be applicable to a group of students and individualized where appropriate. In some cases the student's selected persona can introduce a story line and in other cases several personae are directly involved in an interaction. For example, the story line can be used as the basis for a reward such as "We're off to the beach to play volleyball. Join us when you finish your homework, then later help us solve the mystery of the vanishing lake."

In summary, the use of the student data object in agent behavior processing is illustrated in Table 13.

TABLE 13: STUDENT DATA USE IN AGENT BEHAVIOR

STUDENT DATA	EFFECTS
Student persona choice	Display persona, including type of character, voice inflection, activity rate, and so forth
Student's characteristics, including grade level, language ability, preferences, and so forth	Narrow candidate utterance template and candidate display
Previous student interactions with materials or agent	In alternative embodiments, utterance and display can relate to previous interactions
Previous utterances in this session	Selection of different utterance template and slots
Previous displays in this session	Selection of different display and display objects

Particular data in the student data object has an important role in determining agent display and insuring its appropriateness, variety, and individualization.

5.6.5. Agent Adaptivity

The adaptation of the agent to the student, and the ABI system's appearance as a virtual tutor, emerges from the agent's updating of data in the student data object. In the preferred embodiment, the agent's knowledge of the student is represented by data in the student data object, in particular in the student pedagogic model, which stores general materials independent data describing how the student learns, and in the materials specific student performance, which stores performance data specific to particular assigned materials and courses. As the student interacts with the system for instruction or homework, the agent receives event messages, which describe the student's learning and performance. The agent updates the student data object with data from these messages. As this data is updated, the agent adapts to the student, and thereby the virtual tutor

individualizes to the student. This adaptation is maintained across sessions with this student.

In the preferred embodiment, the data referenced and updated by the agent are averages or weighted moving averages, giving more weight to recent than past behavior. Thus, the pedagogic model includes, for example, data weighted moving averages of the rates that the student learns discrimination of a certain complexity. Materials specific performance includes, for example, weighted moving averages of data on the student's response time and response latency.

In alternative implementations, agent adaptivity to its student can occur differently. In one alternative embodiment, student model data includes not only averages or weighted moving averages, but also data on the statistical distribution of the parameters involved, such as standard deviations. With this data the agent can recognize the current situation as "normal" or "abnormal" for this student and thereby offer appropriate guidance. Further, this statistical data can optionally include correlations between the agent data, such as between various pedagogic parameters and various materials parameters. In this manner, educational situations can be classified more finely than "normal" or "abnormal" into, for example, "abnormally slow on this fluency drill in view of normal progress on other exercises." In a further embodiment, classification of this statistical data can be done by special executable modules in the agent based on, for example, statistical classification schemes or neural networks.

In a further alternative embodiment, agent action processing can be implemented with more complicated techniques. In the preferred embodiment, agent action processing is done by rules without rule propagation. In alternative embodiments, rule propagation and full rule based systems can be used to transform events into actions. This invention is applicable to other techniques from artificial intelligence that can make this transformation such as Bayesian belief networks.

In another alternative embodiment, the student data object has data modeling student interests and preferences. Such a model enables the agent, for example, to monitor school events and suggest those appropriate to the user. This model also enables the agent to provide rewards tailored to individual students, which enhances the system reinforcement and adds to perceived agent persona personality and to virtual tutor individualization. In a preferred embodiment, this model of student interests can be implemented simply as a set of approximately 200 categories, covering interest in several subdivisions of each school subject area, as well as categories related to sports, leisure time and other areas of student interest. Interest in these categories can be entered in several manners. Student use of the encyclopedia tool can be used to determine areas of current interest. Interest can be directly entered by the student, parent, or teacher. Alternately, student interest in materials can inquired for when the materials terminate. After an assignment, the student could provide semiotic feedback by selecting from a row of faces with different expressions. Alternately, the student can be quizzed on interests in a posed branching manner.

6. SPECIFIC EMBODIMENTS, CITATION OF REFERENCES

The present invention is not to be limited in scope by the specific embodiments described herein. Indeed, various modifications of the invention in addition to those described herein will become apparent to those skilled in the art from the foregoing description and accompanying figures. Such modifications are intended to fall within the scope of the appended claims.

Various publications are cited herein, the disclosures of which are incorporated by reference in their entireties.

WHAT IS CLAIMED IS:

1. A method of operating an agent based instruction system for interactive instruction of a student over a plurality of instructional sessions, said method comprising:

(a) presenting interactive instruction to said student by executing one or more materials on a computer accessed by said student for a current instructional session;

(b) monitoring said interactive instruction of said student during said current instructional session;

(c) storing information responsive to said monitoring of said student during said current instructional session for use during subsequent instructional sessions; and

(d) outputting information on said computer to guide said student in said interactive instruction, said output information responsive to said monitoring of said student during said current instructional session and to said stored information responsive to said monitoring of said student during previous instructional sessions;

whereby said system acts as a virtual tutor adapted to said student, said virtual tutor for guiding said interactive instruction of said student.

2. The method according to claim 1 further comprising after said step of monitoring a further step of controlling said one or more materials, said controlling responsive to said monitoring of said student during said current instructional session and to said stored information responsive to said monitoring of said student during previous instructional sessions, and wherein said one or more instructional materials are responsive to said controlling·

whereby said interactive instruction presented by said system is individualized to said student.

3. The method according to claim 2 wherein said step of monitoring further comprises generating monitoring

information and storing said monitoring information in a student data object.

4. The method according to claim 2 wherein said step of monitoring further comprises monitoring pedagogic characteristics of said interactive instruction of said student by said one or more materials, and said step of controlling further comprises controlling said one or more materials in order to present interactive instruction with said pedagogic characteristics.

5. The method according to claim 4 wherein said pedagogic characteristics are selected from the group comprising time pacing of interactive instruction, new concept seeding rate, density of examples, and discrimination difficulty.

6. The method according to claim 2 wherein said step of monitoring further comprises monitoring according to an instructional context and an instructional format adopted by each of said one or more materials.

7. The method according to claim 2 further comprising before said step of controlling and said step of outputting a further step of generating at least one action responsive to said monitoring of said student during said current instructional session and to said stored information responsive to said monitoring of said student during previous instructional sessions, and wherein said step of controlling is responsive to said generated action(s) and wherein said step of outputting is responsive to said generated action(s).

8. The method according to claim 7 wherein said of generating step generates action(s) according to one or more tables of rules.

9. The method according to claim 7 wherein said of generating step generates action(s) according to one or more methods selected from the group of expert systems, neural networks, Bayesian belief networks, and statistical pattern recognition.

10. The method according to claim 1 wherein said step of outputting information further comprises (i) a step of selecting an utterance and an affect from one or more utterance tables in a manner responsive to said monitoring of said student during said current instructional session and to said stored information responsive to said monitoring of said student during previous instructional sessions, and (ii) a step of selecting a visual display from one or more tables of display behaviors in a manner responsive to said utterance, said affect, and said monitoring and said stored information.

11. The method according to claim 10 wherein said step of monitoring further comprises monitoring pedagogic information describing the pedagogic characteristics of said student, wherein said selected display behavior comprises one or more personae, and wherein said selected visual display is further responsive to said pedagogic information;

whereby said one or more personae present a life-like appearance individualized to said student's cognitive style.

12. The method according to claim 1 further comprising prior to said step of outputting a step of inputting at least one request of said student for guidance in said interactive instruction, and wherein said output information is further responsive to said input requests.

13. The method according to claim 1 wherein said computer accessed by said student is one computer of a plurality of computers interconnected by a network, and

wherein said student can access any of said plurality of computers for an instructional session.

14. The method according to claim 1 wherein said output information further comprises an utterance, and wherein the step of outputting further comprises a step of selecting said utterance from one or more candidate utterances, and a further step of outputting said utterance as text or speech.

15. The method according to claim 14 wherein said step of selecting further comprises selecting said one or more candidate utterance from one or more tables of utterances.

16. The method according to claim 15 wherein said one or more table of utterances comprise at least 1,000 utterances.

17. The method according to claim 14 wherein said one or more candidate utterances comprises at least 10 utterances.

18. The method according to claim 1 wherein said responsive output information further comprises a display behavior, and wherein the step of outputting further comprises a step of selecting said display behavior from one or more candidate display behaviors, and a further step of outputting said display behavior.

19. The method according to claim 18 wherein said step of selecting further comprises selecting one or more candidate display behaviors from one or more tables of display behaviors.

20. The method according to claim 19 wherein said one or more tables of display behaviors comprise at least 1,000 display behaviors.

21. The method according to claim 18 wherein said one or more candidate display behaviors comprises at least 10 display behaviors.

22. The method according to claim 18 wherein said step of outputting said selected display behavior comprises outputting one or more modalities selected from the group consisting of text, voice, audio, animation, video, and pre-formatted animated sequences.

23. The method according to claim 18 wherein said selected display behavior comprises one or more persona.

24. The method according to claim 1 wherein said one or more materials is a plurality of materials.

25. The method according to claim 1 wherein said one or more students is a plurality of students.

26. The method according to claim 1 wherein said step of monitoring further comprises monitoring pedagogic information describing the pedagogic characteristics of said student in a manner independent of the subject matters of said one or more materials.

27. The method according to claim 1 wherein said step of monitoring further comprises monitoring progress and performance information describing the progress and performance of said student in said interactive instruction presented by each of said one or more materials.

28. The method according to claim 1 wherein said step of storing further comprises storing said information responsive to said monitoring in one of one or more student data objects, each student data object of said one or more student data objects being uniquely associated with exactly one student of said one or more students.

29. The method according to claim 1 wherein said one or more materials further comprise a material engine and one or more materials data, and wherein said step of presenting interactive instruction by executing one or more materials further comprises executing said materials engine which references said one or more materials data.

30. The method according to claim 29 wherein said materials data further comprise notations, and wherein said step of monitoring further comprises said materials engine referencing said notations in order to generate monitoring information.

31. A method of operating an agent based instruction system for interactive instruction of one or more students, said method comprising:

(a) checking the authority of one of said students to access said system for interactive instruction by one or more materials at a computer, said computer being one computer of a plurality of computers interconnected by a network;

(b) loading to said computer software and data for said interactive instruction;

(c) executing said one or more materials on said computer for presenting interactive instruction to said one student; and

(d) outputting information on said computer to guide said student in said interactive instruction, said output information responsive to performance of said interactive instruction of said student.

32. The method according to claim 31 further comprising after said step of loading a further step of monitoring said interactive instruction, said monitoring generating monitoring information that monitors said interactive instruction of said student, and wherein said step of

outputting information is responsive to said monitoring information;

whereby said system acts as a virtual tutor adapted to said one student, said virtual tutor for guiding said interactive instruction of said student.

33. The method according to claim 31 further comprising after said monitoring step a further step of controlling said one or more materials to present instruction to said student that is responsive to said monitoring information, whereby said interactive instruction presented by said system is individualized to said student.

34. The method according to claim 33 further comprising after said loading step a further step of maintaining a model of said one student, said maintaining responsive to said monitoring information of said student, and wherein the step of outputting and the step of controlling are further responsive to said model of said student, whereby said outputting step and said controlling step adapt to said student thereby individualizing said interactive instruction of said student.

35. The method according to claim 34 wherein said step of maintaining a model of said one student further comprises storing said model in a student data object.

36. The method according to claim 35 wherein said step of checking the authority further checks the authority of said one student to access said student data object.

37. The method according to claim 35 further comprising a step of validating after said step of loading and prior to said step of maintaining, said step of validating validates said computer software loaded for said interactive instruction in order to verify that said loaded computer

software is correctly identified as software that correctly maintains said model.

38. The method according to claim 31 wherein said one or more materials comprise a plurality of materials.

39. The method according to claim 31 wherein one or more of said plurality of computers are configured as one or more server computers for holding databases of software and data, and said step of loading further comprises loading across said network to said computer from said databases.

40. The method according to claim 39 wherein said loading occurs when said software and data is demanded at said computer.

41. The method according to claim 39 further comprising caching read-only data on said computer.

42. The method according to claim 31 wherein said step of outputting information further comprises outputting information in one or more output modalities.

43. The method according to claim 42 where said one or more output modalities are one or more output modalities selected from the group consisting of text, graphics, speech, audio, animation, video, and pre-formatted animated sequences.

44. The method according to claim 42 wherein said step of outputting selects said output modalities to output one or more persona or personae responsive to said interactive instruction.

45. The method according to claim 42 wherein said output modalities further comprise content, and said step of

outputting information further comprises a step of loading said content to said computer.

46. A method of operating an agent based instruction system for interactive instruction of one or more students over a plurality of instructional sessions, said method comprising:

(a) presenting interactive instruction to one of said students by executing one or more materials on a computer accessed by said student for a current instructional session;

(b) monitoring said interactive instruction of said student during said current instructional session;

(c) storing information responsive to said monitoring of said student during said current instructional session for use during subsequent instructional sessions;

(d) determining an affect responsive to said monitoring of said student during said current instructional session and to said stored information responsive to said monitoring of said student during previous instructional sessions; and

(e) outputting information on said computer to guide said student in said interactive instruction, said output information responsive to said affect, to said monitoring of said student during said current instructional session, and to said stored information responsive to said monitoring of said student during previous instructional sessions, said output information comprising a visual display;

whereby said visual display is responsive to said affect and to said interactive instruction in a life-like manner.

47. The method according to claim 46 wherein said step of monitoring further comprises monitoring pedagogic characteristics of said student, and wherein the step of

determining an affect determines an affect further responsive to said pedagogic characteristics;

whereby said visual display is responsive in a life-like manner individualized to said student's cognitive style.

48. The method according to claim 46 wherein said output information further comprises an utterance, and wherein the step of outputting further comprises a step of selecting said utterance from one or more tables of utterances in a manner responsive to said monitoring and to said stored information, and a further step of outputting said utterance as text or speech;

whereby said utterance and said visual display are responsive in a life-like manner.

49. The method according to claim 46 wherein said step of outputting further comprises selecting said visual display from one or more tables of visual displays in a manner responsive to said affect, to said monitoring, and to said stored information.

50. The method according to claim 46 wherein said selected display behavior comprises one or more persona.

51. An agent based instruction system for interactive instruction of one or more students, said system comprising:

(a) one or more computers having interactive input/output devices and interconnected by a network;

(b) one or more materials executable on said one or more computers, each said material for presenting interactive instruction to said one or more students and for generating monitoring information that monitors said interactive instruction; and

(c) one or more agents executable on said one or more computers, each said agent associated with exactly one

of said students and each said student associated with exactly one of said agents, each said agent comprising

(i) action processing for controlling said one or more materials to instruct said associated student, said controlling being responsive to said monitoring information that monitors said interactive instruction of said associated student, and

(ii) behavior processing for outputting information to guide said associated student, said outputting being responsive to said monitoring information that monitors said interactive instruction of said associated student;

whereby said system acts as a virtual tutor to each of said students and said interactive instruction of each of said students is individualized to each student.

52. The system according to claim 51 further comprising executive software for interfacing said one or more materials and said one or more agents to said one or more computers and to said network.

53. The system according to claim 51 wherein said network is configured to permit any one of said students to access any one of said one or more materials from any one of said computers.

54. The system according to claim 51 wherein said network is configured to permit one or more of said computers to be located in one or more locations.

55. The system according to claim 54 wherein one or more of said locations are residences of one or more of said students.

56. The system according to claim 54 wherein one or more of said materials presents homework to one or more of said students.

57. The system according to claim 51 wherein said network is further configured to be a packet switched network.

58. The system according to claim 51 wherein said agent action processing is further responsive to a request for guidance from said associated student, and wherein said agent behavior processing is further responsive to said request for guidance.

59. The system according to claim 58 wherein said agent behavior processing further comprises (i) utterance generation processing for selecting an utterance and an affect responsive to said monitoring information or to said request for guidance, and (ii) visual display generation for selecting a visual display responsive to said utterance and said affect.

60. The system according to claim 59 wherein said agent behavior processing selects said utterance and said visual display to represent a persona, said selection being from a table of available personas and according to the preferences of said associated student.

61. The system according to claim 51 wherein said materials further comprise instructional materials executable on said computers for interactive instruction of said students, and tools executable on said computers for assistance of said students in said interactive instruction.

62. The system according to claim 61 wherein said instructional materials further comprise instructional materials appropriate to primary or secondary education.

63. The system according to claim 61 wherein said tools further comprise one or more tools selected from the group

comprising a calculator, a dictionary, a thesaurus, an atlas, an encyclopedia, and a network search tool.

64. The system according to claim 61 wherein said tools further comprise a starfish tool for displaying and selecting relationships.

65. The system according to claim 51 further comprising one or more schedule/calendar tools executable on said computers, each said schedule/calendar tool associated with exactly one of said plurality of students and each said student associated with exactly one of said schedule/calendar tools, each said schedule/calendar tools for outputting to said associated student information relating to scheduled activities of said associated student, and for providing to said agent of said associated student information relating to scheduled activities of said associated student.

66. The system according to claim 65 further comprising data areas for each of said students characterizing scheduled activities according to deadline date and priority, and wherein said schedule/calendar tool further limits said associated student to interactive instruction according to said data areas characterizing said scheduled activities.

67. The system according to claim 51 further comprising one or more communication tools executable on said computers for providing forms of interactive group instruction to a group of said students, and for generating monitoring information that monitors said group instruction of each student in said group.

68. The system according to claim 67 where said forms of group instruction are selected from the group consisting of exchange of messages, group work on a shared material, and group participation in educational contests.

69. The system according to claim 51 wherein one or more of said materials further comprises materials engine software and materials data, and wherein said materials engine process said materials data to present said interactive instruction.

70. The system according to claim 69 wherein each of said materials data further comprises:

(a) a plurality of display objects for presentation;

(b) sequencing logic for controlling the order of said presentation of said plurality of display objects; and

(c) notations for causing generation of said monitoring information.

71. The system according to claim 51 wherein one or more of said materials comprises a program having data and instructions.

72. The system according to claim 51 further comprising pedagogic information data areas for each student, said pedagogic information data areas comprising data for a pedagogic model of said student, and wherein said agent action processing and said agent behavior processing is further responsive to said pedagogic information data areas.

73. The system according to claim 72 wherein said data in said pedagogic information data areas does not depend on the subject matter of said one or more materials.

74. The system according to claim 72 wherein said agent action processing updates said pedagogic information data areas associated with said agent's associated student, said updating being responsive to said monitoring information, whereby said associated agent adapts to said student.

75. The system according to claim 72 further comprising progress and performance data areas for each student, said progress and performance data areas comprising data describing progress and performance of each said student in said interactive instruction, and wherein said agent action processing and said agent behavior processing are further responsive to said progress and performance information data areas.

76. The system according to claim 75 wherein said agent action processing updates said progress and performance data areas associated with said associated student of said agent, said updating responsive to said monitoring information;

whereby said associated agent adapts to said student.

77. The system according to claim 51 wherein one or more of said materials presents said interactive instruction according to an education paradigm, and wherein said monitoring information generated by said materials further comprises pedagogic information classified according to said education paradigm of said materials.

78. The system according to claim 77 further comprising control parameters for each material that have an educational paradigm, wherein each material presents said interactive instruction according to values of said control parameters, and wherein said agent action processing assigns said values of said control parameters;

whereby said agent action processing controls said materials.

79. The system according to claim 77 wherein said educational paradigm is selected from the group consisting of interactive tutorial, fluency exercise, paired association exercise, discrimination formation exercise, and simulation exercise.

80. The system according to claim 77 wherein said educational paradigm is standardized according to an instructional context and an instructional format.

81. The system according to claim 80 wherein said instructional context is selected from the group consisting of prerequisite test, prerequisite review, pretest, new material introduction, new material discrimination, new material review, fluency exercise practice, review practice, and unit mastery test.

82. The system according to claim 80 wherein said instructional format is selected from the group consisting of multiple choice, unprompted fill-in-the-blank, fill-in-the-blank selected from a list, paired associates by letter, paired associates by dragging, paired associates by matching, computation, simulation to identify parts of figures or text, and simulation game.

83. The system according to claim 51 wherein said agent action processing generates at least one action responsive to said monitoring information.

84. The system according to claim 83 wherein said agent action processing further comprises software using one or more techniques selected from the group of expert systems, neural networks, Bayesian belief networks, and statistical pattern recognition.

85. The system according to claim 83 further comprising a plurality of tables of rules, and wherein said agent action processing software further comprises software referencing said plurality of tables of rules in order to control said one or more materials and to generate said action(s).

86. The system according to claim 85 wherein said plurality of tables of rules comprises a policy filter table,

a decision weight table, and a selection criteria table, and wherein said agent action processing references said policy filter table for determining one or more candidate actions, references said decision weight table for ranking said one or more candidate actions, and references said selection criteria table for selecting said action(s) from said ranked candidate actions.

87. The system according to claim 83 wherein said agent behavior processing is responsive to said action(s) in order to output information responsive to said monitoring information.

88. The system according to claim 87 further comprising a plurality of tables of possible outputs, and wherein said agent behavior processing software references said tables of possible outputs.

89. The system according to claim 88 wherein said one or more tables of possible outputs further comprise an utterance template table and a display behavior table, and wherein said agent behavior processing references said utterance template table for selecting an utterance and an affect according to said action(s), and references said display behavior table for selecting a visual display according to said action(s), said utterance, and said affect.

90. The system according to claim 87 wherein said agent behavior processing outputs information in a plurality of output modalities.

91. The system according to claim 87 wherein said output modalities are one or more output modalities selected from the group consisting of text, graphics, speech, audio, animation, video, and pre-formatted animated sequences.

92. The system according to claim 51 further comprising one or more student data objects, each of said student data objects associated with exactly one student, and wherein each agent stores information reflecting said monitoring information of said one student associated with said agent in said one student data object associated with said associated student.

93. The system according to claim 92 wherein one or more of said computers are configured as server systems, and wherein said student data objects are stored on said server systems.

94. The system according to claim 92 further comprising one or more portable media, each said portable media for storing one student data object associated with one of said students, and wherein said one or more computers are further capable of accessing said student data object on said portable media.

95. The system according to claim 94 wherein said portable medium for storing said student data object is physically possessed by said one associated student.

96. The system according to claim 94 wherein said portable media further comprises a portable hard disk or a portable floppy disk.

97. The system according to claim 96 wherein said student data object is stored in an encrypted fashion on said portable storage media.

98. The system according to claim 94 wherein said portable media further comprises a portable card which comprises:

(a) a stable storage for storing said student data object; and

(b) a processor for responding to calls to access said student data object from said one or more computers.

99. The system according to claim 98 wherein said portable media is a smart-card.

100. The system according to claim 92 wherein said student data object associated with one student further comprises:

(a) pedagogic information describing a pedagogic model of said one student; and

(b) progress and performance information for describing the progress and performance of said student in said materials.

101. The system according to claim 100 wherein said agent associated with each student further comprises agent software and said student data object associated with said student, and wherein said agent software references and updates said associated student data object.

102. The system according to claim 51 wherein one or more of said computers are configured as server systems, wherein said server systems store said one or more materials and one or more agents, and wherein said one or more computers downloads said materials and said one or more agents from said one or more server systems across said computer network.

103. The system according to claim 102 further comprising databases of pedagogic information and materials progress and performance information for said one or more students, and wherein said databases are stored on said server systems.

104. The system according to claim 103 further comprising reporting software for generating reports from

said databases of pedagogic information and materials progress and performance information for said one or more students.

105. The system according to claim 51 further comprising one or more system managers executable on said one or more computers, wherein said interactive input/output devices include a computer display at each of said one or more computers, and wherein said system manager executing on one computer partitions said display of said one computer into one or more display areas.

106. The system according to claim 105 wherein said one or more display areas are selected from the group comprising an agent area for agent interactivity, a materials area for materials interactivity, and a system area in which said system manager displays selection icons for available system tools and facilities.

107. The system according to claim 106 wherein said one or more materials and said one or more agents provide facilities always available to said one or more students and said materials area and said agent area comprise sub-areas whose selection activates one of said always available facilities.

108. The system according to claim 106 wherein said one or more display areas change in size from time to time.

109. The system according to claim 51 wherein one or more teachers directs instruction of one or more of said students, said system further comprising:

(a) one or more teacher materials executable on said one or more computers, each said teacher material for presenting interactive instruction to teachers in the use of said system and in the use of materials directed to said

students and for generating monitoring information that monitors said interactive instruction; and

(b) one or more teacher agents executable on said one or more computers, each said teacher agent associated with exactly one of said teachers and each said teacher associated with exactly one of said agents, each said agent comprising

(i) action processing for controlling said one or more materials to instruct said associated teacher, said controlling being responsive to said monitoring information that monitors said interactive instruction of said associated teacher, and

(ii) behavior processing for outputting information to guide said associated teacher, said outputting being responsive to said monitoring information that monitors said interactive instruction of said associated teacher.

110. An agent based instruction system for interactive instruction of one or more students, said system comprising:

(a) one or more computers having interactive input/output devices and interconnected by a network;

(b) one or more materials executable on said one or more computers, each said material for presenting interactive instruction to said one or more students and for generating monitoring information that monitors said interactive instruction; and

(c) one or more agents executable on said one or more computers, each said agent associated with exactly one of said students and each said student associated with exactly one or said agents, each said agent comprising

(i) utterance generation processing for selecting an affect and an utterance to guide said one associated student, said selecting of said affect and said utterance being responsive to said monitoring information that monitors said interactive instruction of said one associated student, and

(ii) visual display generation for selecting a visual display to guide said one associated student, said selecting of said visual display being responsive to said affect, to said utterance, and to said monitoring information that monitors said interactive instruction of said associated student, and

(iii) output processing for outputting to said associated student said selected utterance and said selected visual display;

whereby said selected visual display and said selected utterance are responsive to said affect and to said interactive instruction in a life-like manner.

111. The system according to claim 110 wherein said monitoring information further comprises pedagogic information describing the pedagogic characteristics of said associated student, and wherein utterance generation processing selects an affect further responsive to said pedagogic information;

whereby said selected visual display and said selected utterance are further responsive in a manner individualized to a cognitive style of said associated student.

112. The system according to claim 110 wherein said utterance generation processing selects said affect and said utterance from one or more tables of utterances.

113. The system according to claim 110 wherein said visual display processing selects said visual display from one or more tables of visual displays.

114. The system according to claim 110 wherein said selected visual display comprises one or more persona.

115. The system according to claim 110 further comprising action processing for generating at least one

action and for controlling said one or more materials to instruct said associated student, said generating and said controlling being responsive to said monitoring information that monitors said interactive instruction of said associated student, and wherein said utterance generation processing and said visual display selection processing are further responsive to said action(s).

116. A method of operating an agent based instruction system for instruction of a plurality of students, said method comprising:

(a) executing one or more materials on a computer for presenting interactive instruction to one student of said plurality of students, said computer being one computer of a plurality of computers interconnected by a network, each computer of said plurality having interactive input/output devices;

(b) generating monitoring information that monitors said interactive instruction presented to said student; and

(c) controlling said one or more materials to instruct said student, said controlling being responsive to said information monitoring the interactive instruction of said student;

whereby said interactive instruction is individualized to each student of said plurality of students.

117. The method according to claim 116 further comprising after said step of generating a further step of outputting information on said computer to guide said student in said interactive instruction, said outputting information responsive to said monitoring information of said interactive instruction of said student;

whereby said system acts as a virtual tutor adapted to said student, said virtual tutor for guiding said interactive instruction of said student.

118. The method according to claim 117 wherein said output information further comprises one or more persona.

119. The method according to claim 117 wherein said step of outputting information further comprises outputting information in a plurality of output modalities.

120. The method according to claim 119 wherein said output modalities are one or more output modalities selected from the group consisting of text, graphics, speech, audio, animation, video, and pre-formatted animated sequences

121. The method according to claim 117 wherein said step of outputting information further comprises (i) a step of selecting an utterance and an affect from one or more utterance tables in a manner responsive to said controlling step and (ii) a step of selecting a visual display from one or more tables of display behaviors in a manner responsive to said utterance, said affect, and said controlling step.

122. The method according to claim 117 further comprising prior to said outputting step a further step of inputting student requests for guidance in said interactive instruction, and wherein said step of outputting information is further responsive to said student requests.

123. The method according to claim 116 wherein said one or more materials is a plurality of materials.

124. The method according to claim 116 further comprising prior to said executing step a further step of accessing any one of said computers by said one student, said accessing comprising verifying the authority of said one student to access said system for interactive instruction by said one or more materials.

125. The method according to claim 124 wherein one or more of said computers are located in a plurality of locations, and wherein the step of accessing further comprises accessing any of said computers in plurality of locations.

126. The method according to claim 125 wherein one or more of said plurality of locations are residences of one or more of said students, and wherein said one or more materials presents homework in said one or more residences.

127. The method according to claim 116 wherein said step of controlling further comprises controlling according to one or more tables of rules.

128. The method according to claim 127 wherein said one or more tables of rules further comprise a policy filter table, a decision weight table, and a selection criteria table, and wherein said controlling step controls said one or more materials according to one or more determined controlling actions, and wherein said controlling step references said policy filter table for determining one or more candidate controlling actions, references said decision weight table for ranking said one or more candidate controlling actions, and references said selection criteria table for selecting one or more controlling actions from said ranked candidate controlling actions.

129. The method according to claim 116 wherein said step of controlling further comprises controlling according to one or more methods selected from the group consisting of expert systems, neural networks, Bayesian belief networks, and statistical pattern recognition.

130. The method according to claim 116 wherein the step of executing further comprises executing one or more of said one or more materials according to an educational paradigm,

and wherein said monitoring information further comprises pedagogic information classified according to said educational paradigm adopted by said one or more materials.

131. The method according to claim 130 wherein said educational paradigm is standardized according to an instructional context and an instructional format.

132. The method according to claim 116 further comprising after said generating step a further step of updating one student model of a plurality of student models, each student of said plurality of students being associated with exactly one student model, said updating being responsive to the information monitoring the interactive instruction of said student, and wherein the step of controlling said one or more materials to instruct said student is further responsive to said student model of said student;

whereby said one or more materials are individualized to said student.

133. The method according to claim 132 wherein said step of updating said one student model further comprises updating one of a plurality of student data objects, each said student data object for storing one of said models.

134. The method according to claim 133 further comprising, prior to said step of executing, a step of storing said student data objects on a plurality of portable storage media, said portable storage being accessible for said step of updating, each said portable storage media for storing one of said student data objects.

135. The method according to claim 132 wherein said step of updating further comprises updating pedagogic information in said student model, said pedagogic information describes pedagogic characteristics of said student of said plurality

of students in a manner independent of the subject matters of said one or more materials.

136. The method according to claim 132 wherein said step of updating further comprises updating progress and performance information in said student model, said progress and performance information describes the progress and performance of said student in said interactive instruction presented by each of said one or more materials.

137. The method according to claim 116 wherein said one or more materials further comprises one or more instructional materials for presenting interactive instruction to said plurality of students, and one or more tools for presenting interactive assistance to said plurality of students during said interactive instruction.

138. The method according to claim 116 further comprising prior to said controlling step (i) a step of executing one scheduler/calendar tools of a plurality of scheduler/calendar tools, each student of said plurality of students being associated with exactly one scheduler/calendar tool, said one scheduler/calendar tool providing said student with information relating to scheduled activities of said student, and (ii) a step of generating monitoring information that monitors said scheduled activities of said student.

139. The method according to claim 138 wherein the step of executing executes one or more materials that relate to said scheduled activities of said student.

140. The method according to claim 116 further comprising prior to said controlling step (i) a step of executing one or more communication tools for presenting one or more forms of group instruction to a group of students, said group including said student, and (ii) a step of

generating monitoring information that monitors said group instruction of said student.

141. A method of operating an agent based instruction system for instruction of a plurality of students, said method comprising:

(a) a step for executing one or more materials on a computer for presenting interactive instruction to each student, said computer being one computer of a plurality of computers interconnected by a network, each computer of said plurality having interactive input/output devices;

(b) a step for generating monitoring information that monitors said interactive instruction presented to each student of said plurality of students;

(c) a step for controlling said one or more materials to instruct each student, said controlling being responsive to said information monitoring the interactive instruction of each student; and

(d) a step of outputting information to each student in order to guide each student in said interactive instruction, said outputting step responsive to said information monitoring the interactive instruction of each student and responsive to said controlling step;

whereby said interactive instruction is individualized to each student of said plurality of students and acts as a virtual tutor individualized to each student.

142. An agent based instruction system for instruction of a plurality of students, said system comprising:

(a) means for presenting interactive instruction to each student by one or more materials;

(b) means for generating monitoring information that monitors said interactive instruction presented to each student;

(c) means for controlling said one or more materials to instruct each student, said controlling being

responsive to said information monitoring the interactive instruction of each student; and

(d) means for outputting information to each student in order to guide each student in said interactive instruction, said means for outputting being responsive to said information monitoring the interactive instruction of each student and responsive to said means for controlling;

whereby said interactive instruction is individualized to each student and said acts as a virtual tutor individualized to each student.

143. An agent based instruction system for instruction of a student, said system comprising:

(a) one or more materials executable on a computer for presenting interactive instruction to said student, said computer having interactive input/output devices; and

(b) an agent executable on said computer, said agent (i) receiving monitoring information from each of said one or more materials that monitors said interactive instruction of said student, (ii) controlling said one or more materials to instruct said student, said controlling being responsive to said monitoring information, and (iii) outputting information to guide said student, said outputting being responsive to said monitoring information.

144. A system for agent-based, interactive instruction of one or more students over a plurality of instructional sessions, said system comprising:

(a) means for presenting interactive instruction to one of said students by executing one or more materials on a computer accessed by said student for a current instructional session;

(b) means for monitoring said interactive instruction of said student during said current instructional session;

(c) means for storing information responsive to said monitoring of said student during said current

instructional session for use during subsequent instructional sessions; and

(d) means for outputting information on said computer to guide said student in said interactive instruction, said output information responsive to said monitoring of said student during said current instructional session and to said stored information responsive to said monitoring of said student during previous instructional sessions;

whereby said system acts as a virtual tutor adapted to said student, said virtual tutor for guiding said interactive instruction of said student.

145. A system for agent-based, interactive instruction of one or more students over a plurality of instructional sessions, said system comprising:

(a) means for presenting interactive instruction to one of said students by executing one or more materials on a computer accessed by said student for a current instructional session;

(b) means for monitoring said interactive instruction of said student during said current instructional session;

(c) means for storing information responsive to said monitoring of said student during said current instructional session for use during subsequent instructional sessions;

(d) means for determining an affect responsive to said monitoring of said student during said current instructional session and to said stored information responsive to said monitoring of said student during previous instructional sessions; and

(e) means for outputting information on said computer to guide said student in said interactive instruction, said output information responsive to said affect, to said monitoring of said student during said current instructional session, and to said stored information

responsive to said monitoring of said student during previous instructional sessions, said output information comprising a visual display;

whereby said visual display is responsive to said affect and to said interactive instruction in a life-like manner.

146. A computer readable medium comprising instructions for performing the method of claim 1.

147. A computer readable medium comprising instructions for performing the method of claim 30.

148. A computer readable medium comprising instructions for performing the method of claim 42.

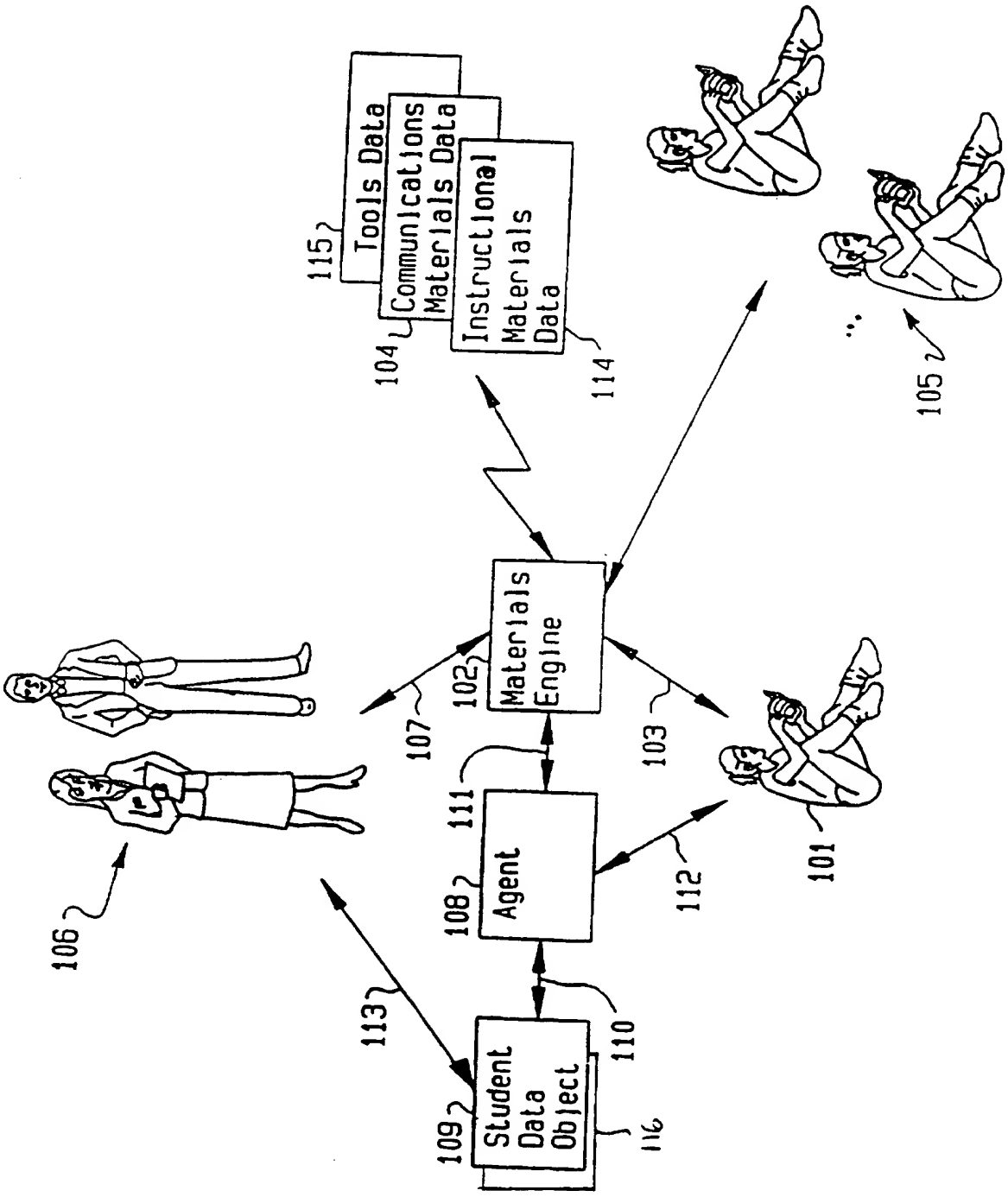

FIG. 1

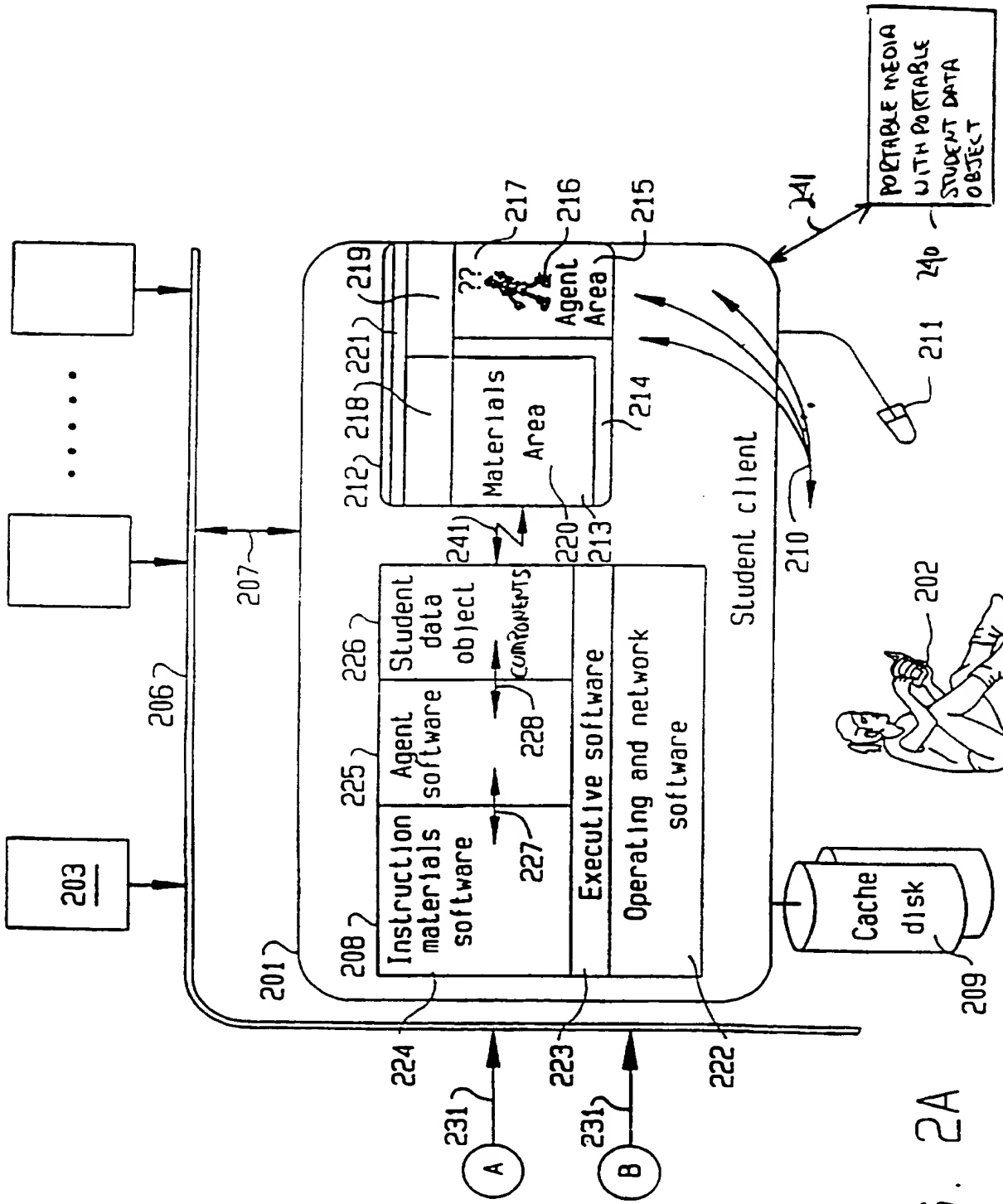

FIG. 2A

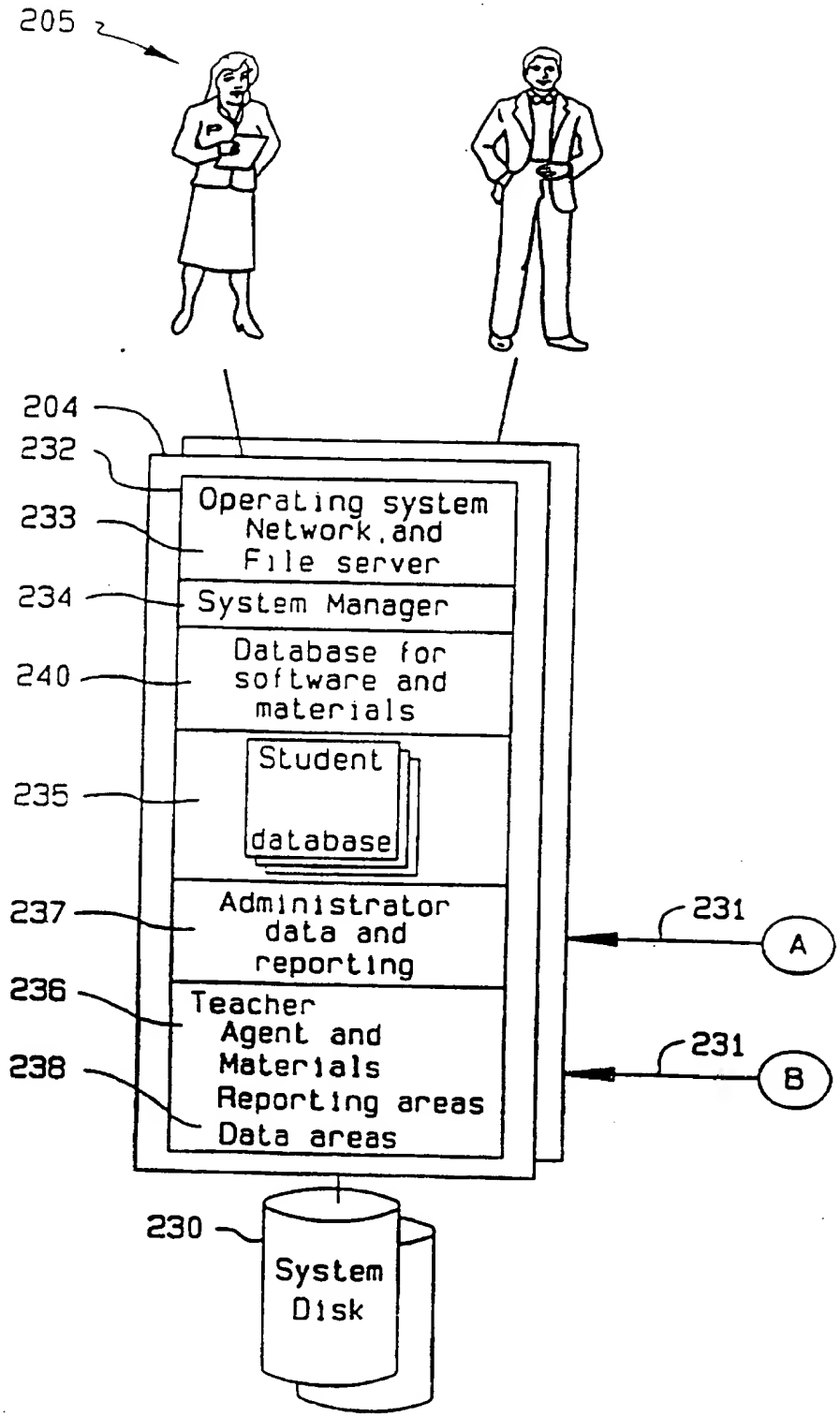

FIG. 2B

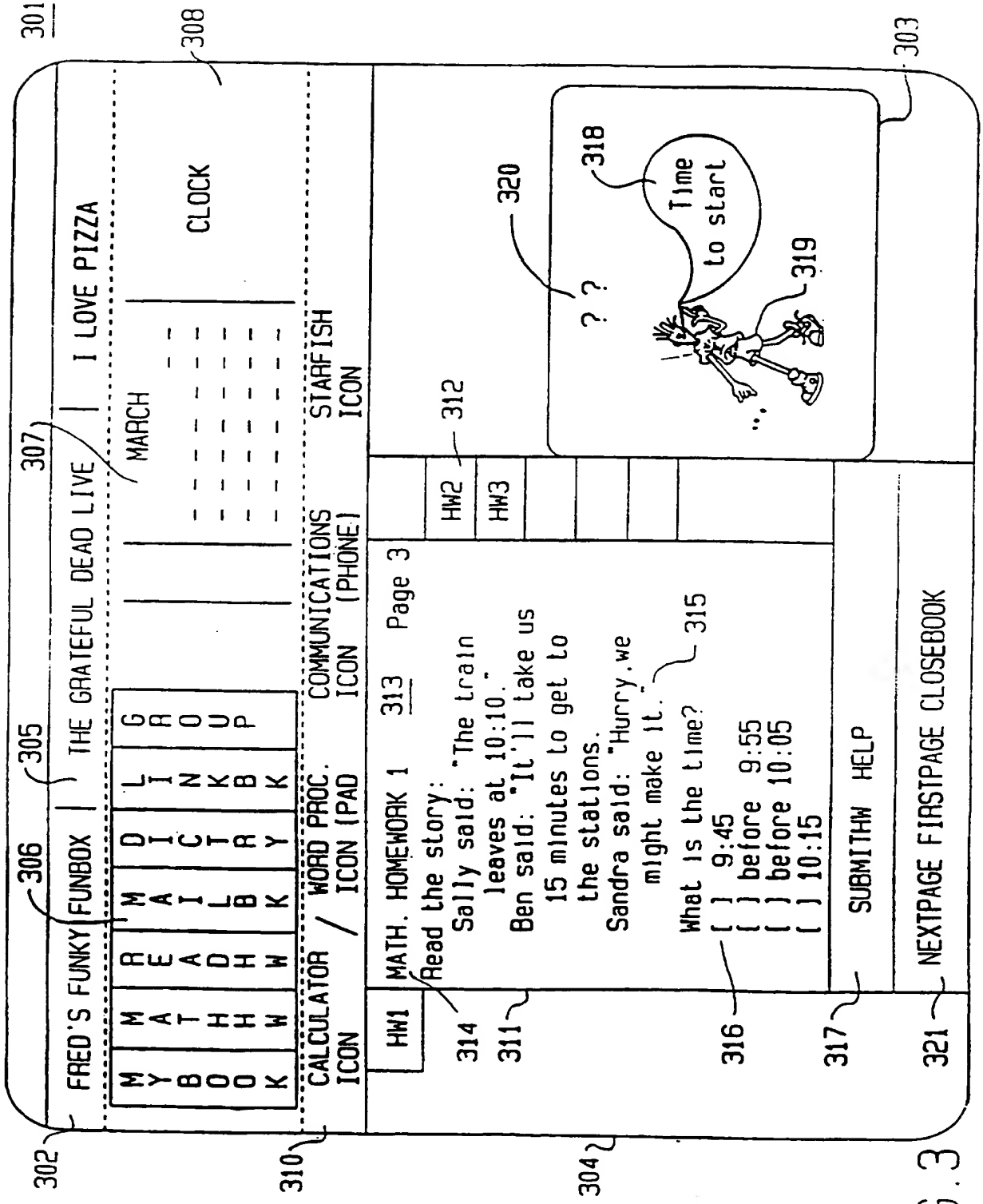

FIG. 3

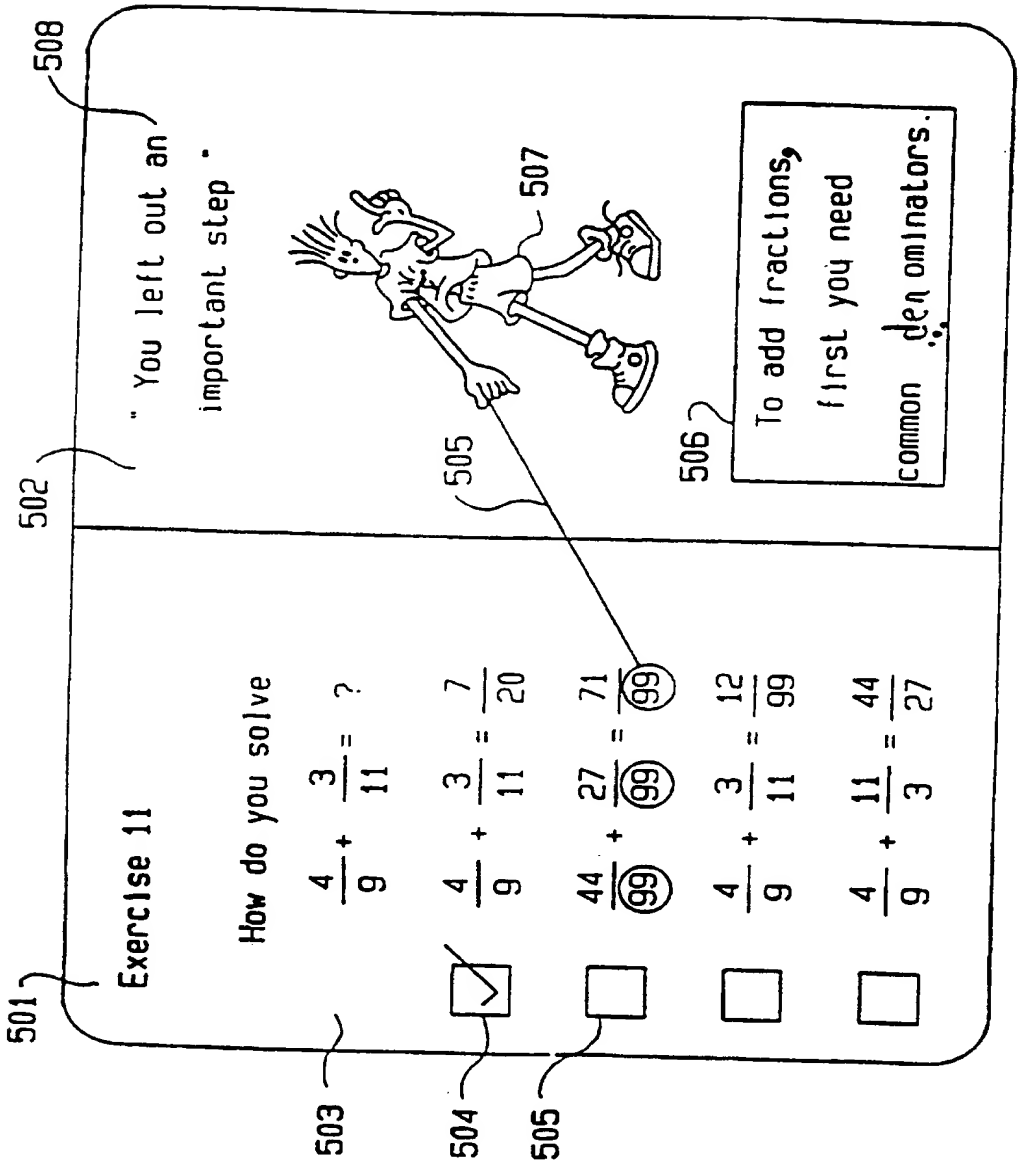

FIG. 4

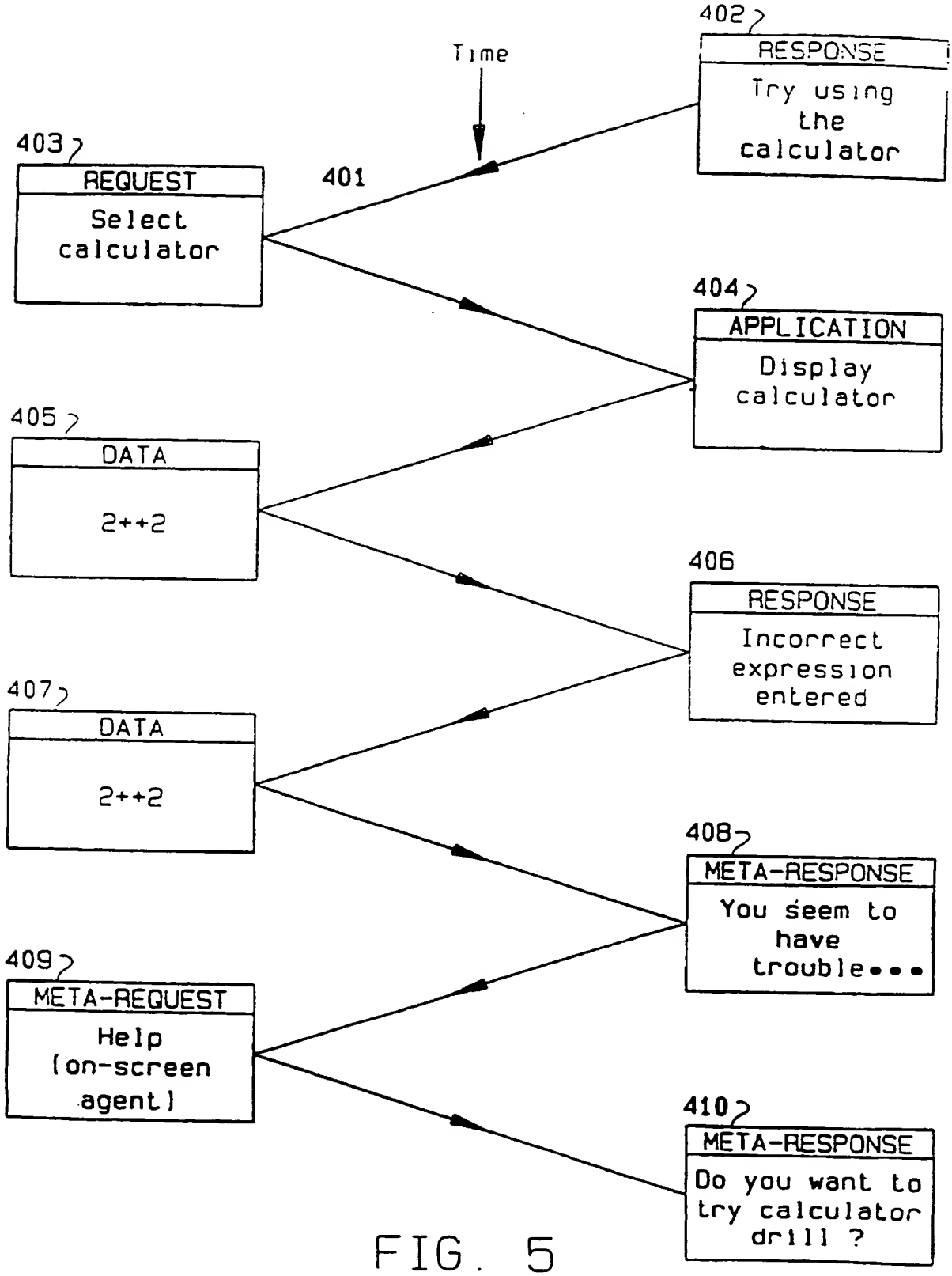

FIG. 5

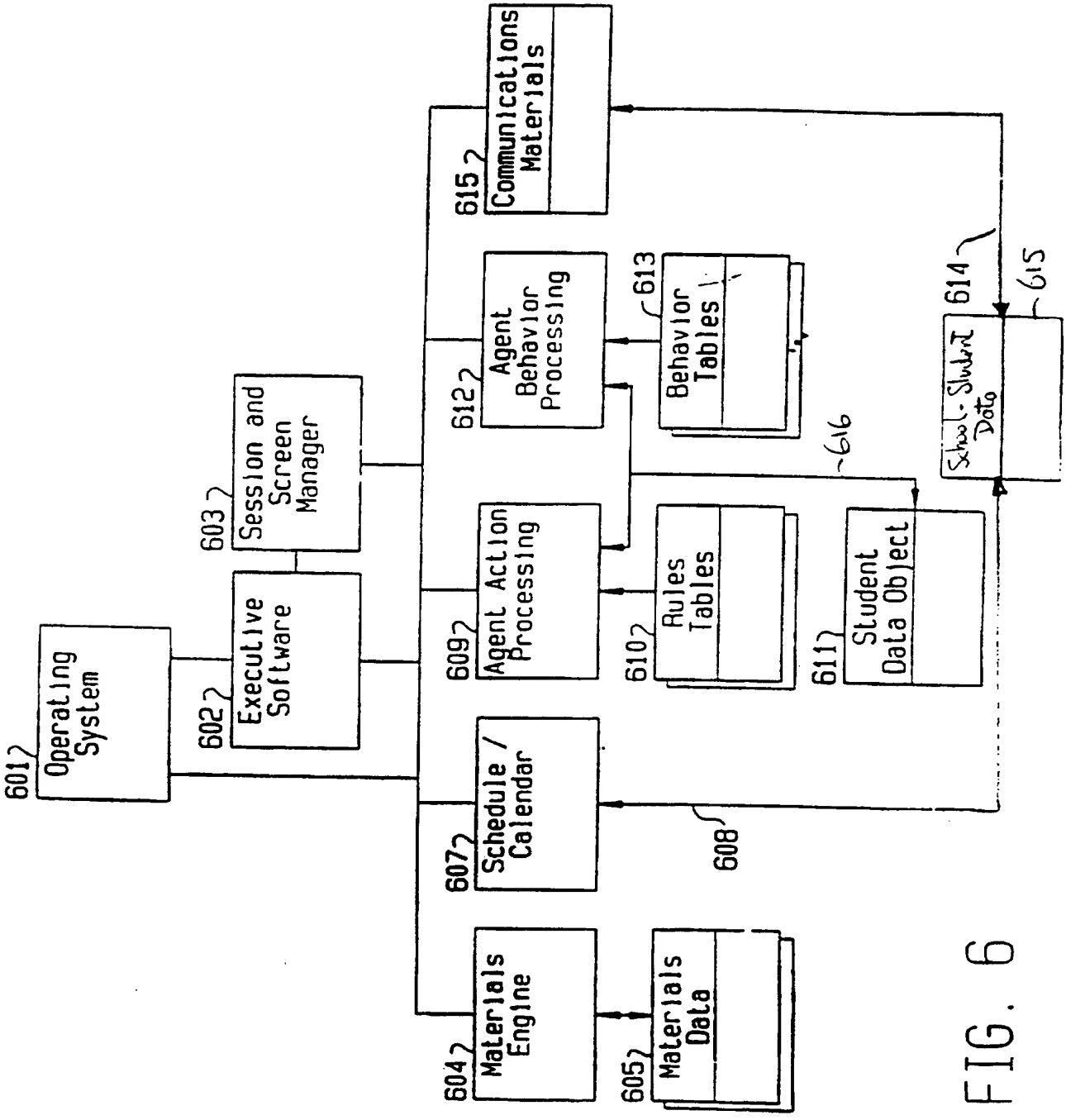

FIG. 6

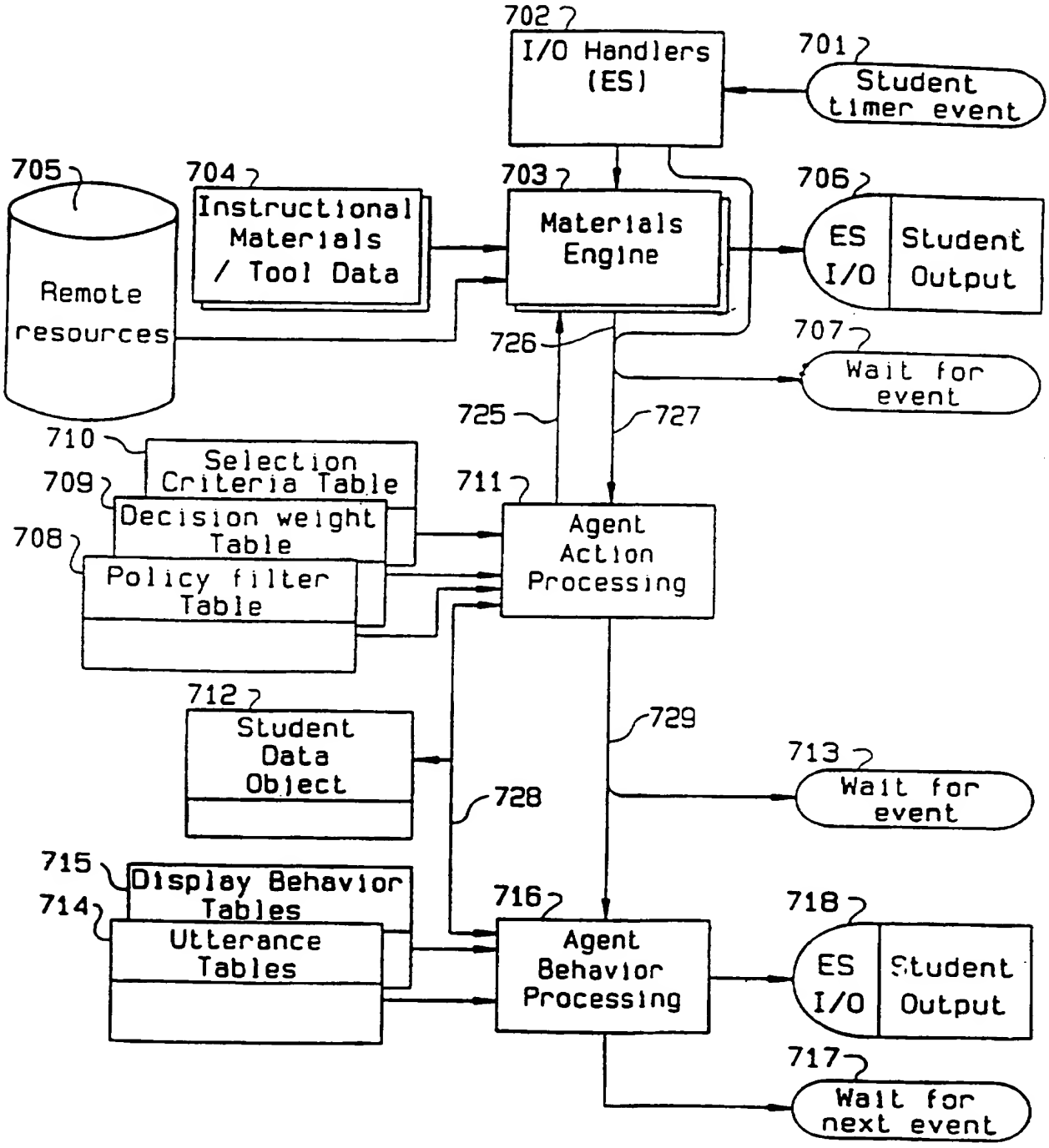

FIG. 7

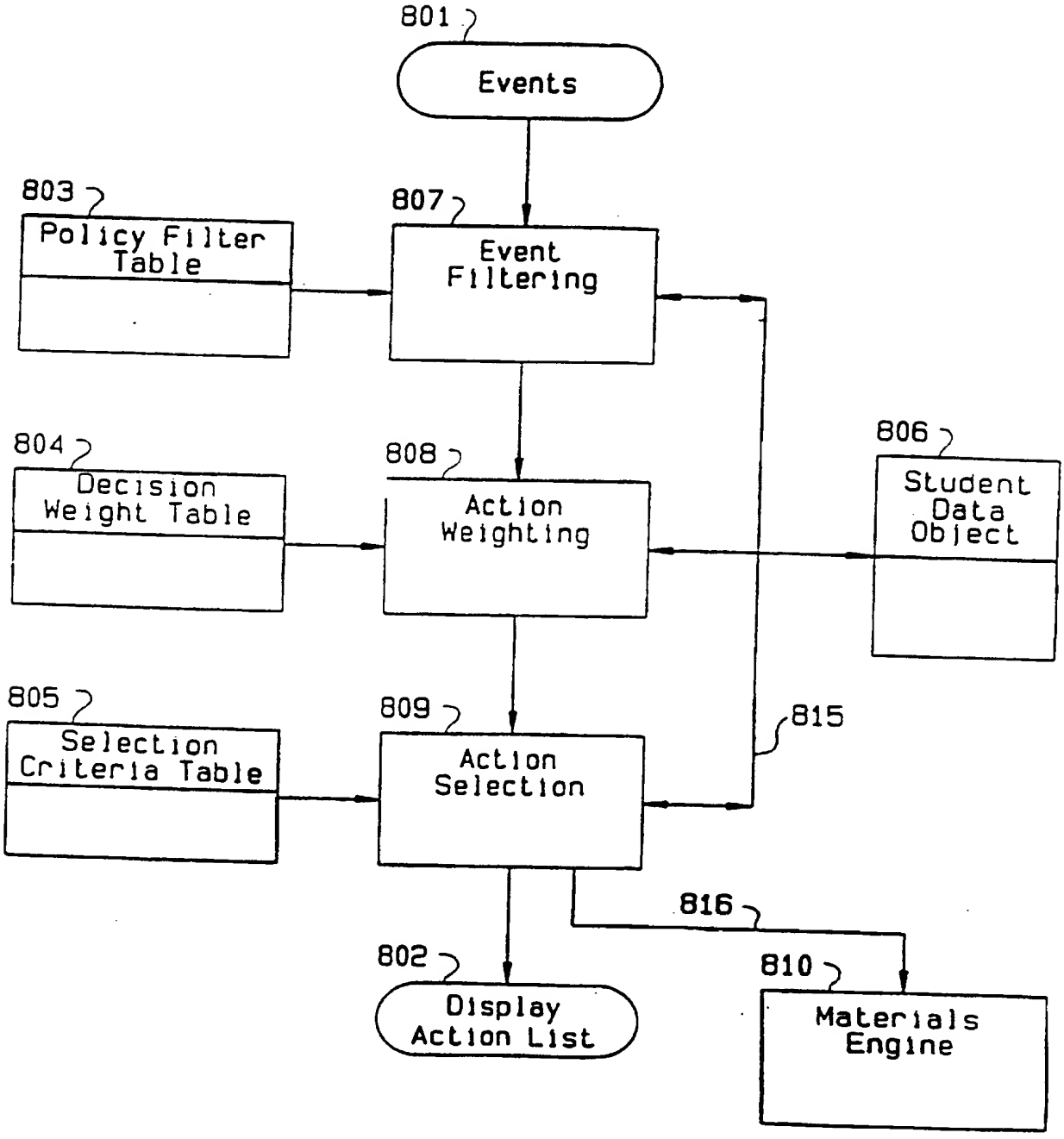

FIG. 8

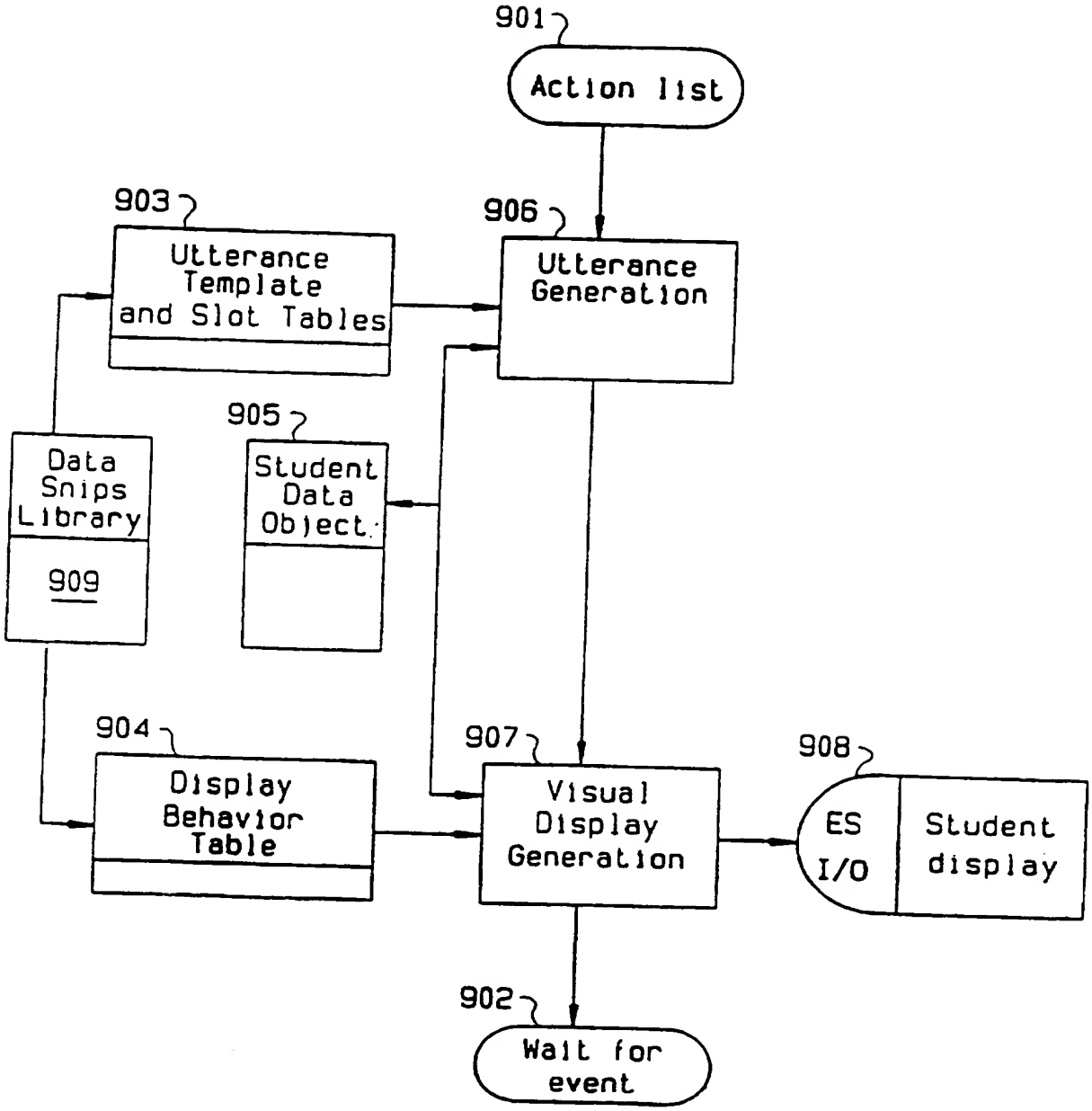

FIG. 9

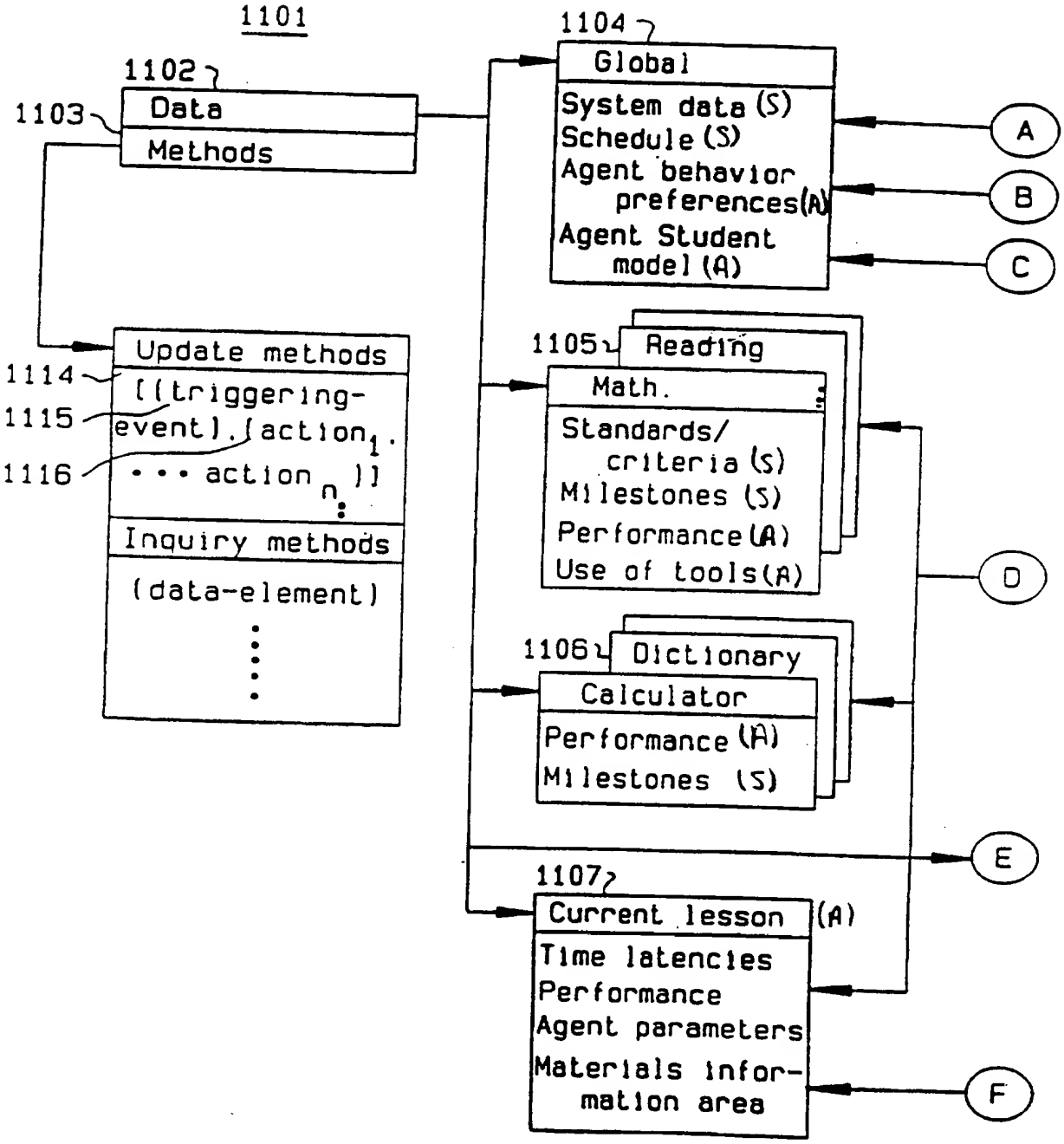

FIG. 10A

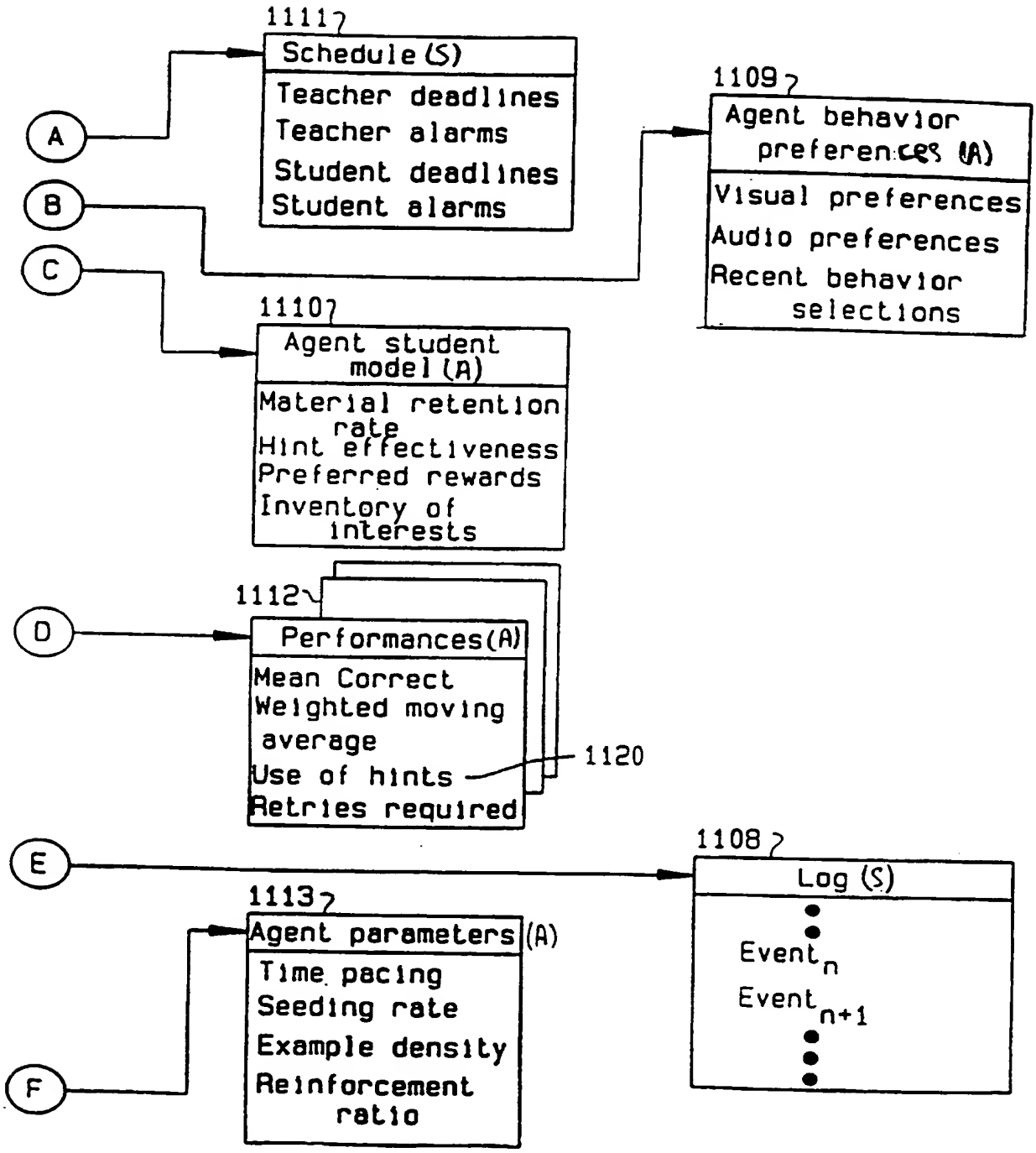

FIG. 10B

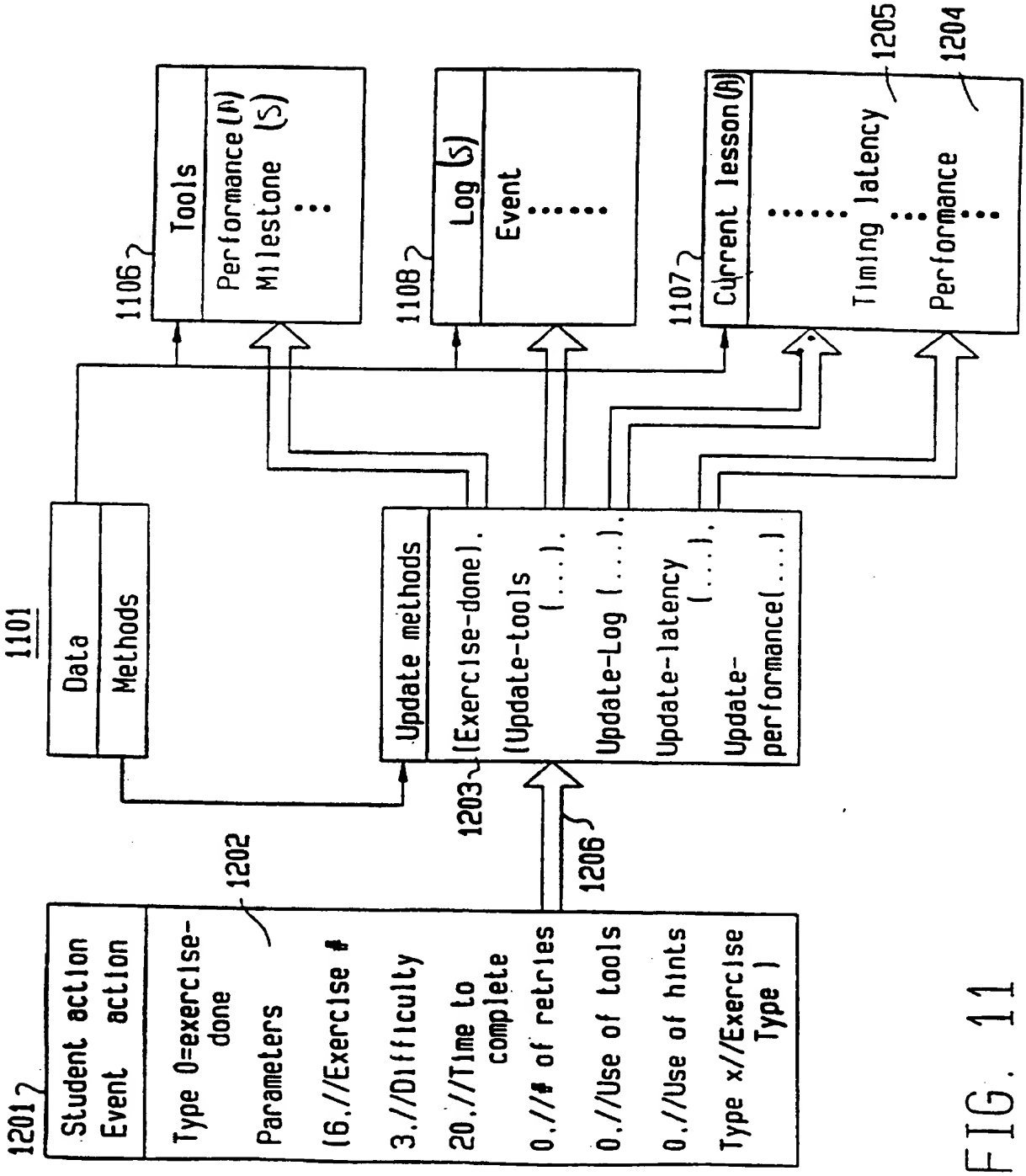

FIG. 11

METAREQUEST METARESPONSE

"Where am I"

"You are one quarter through the lesson on adding fractions."

Student clicks on "More" icon

* MATH UNIT 5, LESSON 6*

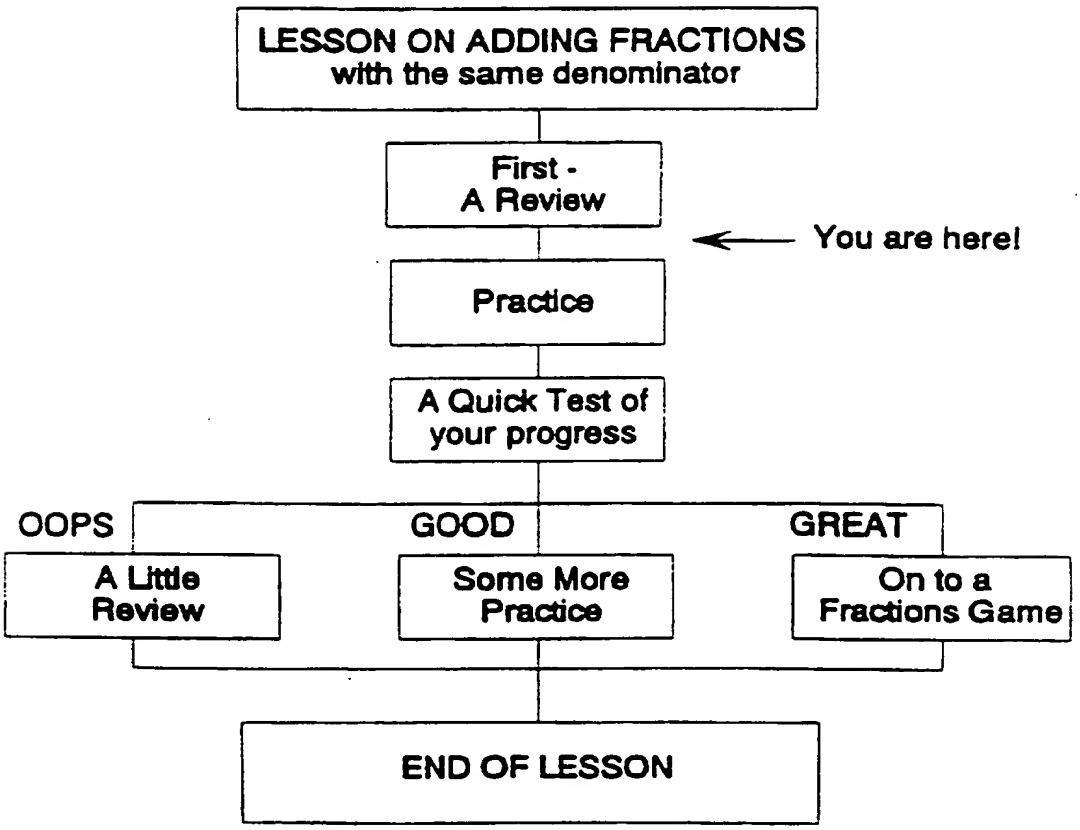

Fig. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/08685

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G09B 5/12
US CL :434/350
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 434/350

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- A	US 5,310,349 A (DANIELS et al) 10 May 1994, col. 3, lines 1-40.	1, 12, 13, 24-28, 31-45, 116-120, 122-127, 129-140, 146-148 --- 2-11, 14-23, 29-30, 46-115, 121, 128, 141-145

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 19 AUGUST 1997	Date of mailing of the international search report 05 SEP 1997
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer WILLIAM H. GRIEB Telephone No. (703) 308-3549

Form PCT/ISA/210 (second sheet)(July 1992)*